MOTOROLA

GETTING ABOARD THE 488-1975 BUS

IMPLEMENTATION OF THE IEEE 488-1975 INSTRUMENTATION BUS WITH THE MC68488 INTERFACE ADAPTER

MOTOROLA MICROCOMPUTERS — SYSTEMS ON SILICON

MOTOROLA

GETTING ABOARD THE 488-1975 BUS

Circuit diagrams external to Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.

FOREWORD

The adoption of IEEE Standard 488-1975 is a significant step toward creation of a universal instrumentation systems approach. This standard allows easy systems configuration involving instruments produced by different manufacturers. As such it greatly simplifies the design of production testing, systems control or scientific data recording systems.

Quite naturally, the 488-1975 Standard has generated a tremendous amount of interest among both instrument and computer manufacturers and measurement equipment users. Among the restraining factors on 488-1975 implementation have been the complexity of the logic protocol and the inavailability of low-cost IC's to satisfy the required logic control needs.

This brochure briefly describes the 488-1975 system architecture and highlights two IC's for system implementation. It is by no means a complete tutorial covering of the 488 System and, therefore, it supplements the IEEE document rather than replacing it. It will be assumed that the reader has access to the IEEE Standard 488-1975 for technical reference.

Since this document is primarily intended as a reference piece, liberal use of tables, illustrations and diagrams has been made to present the topic in a cohesive manner.

Portions of this document have been taken from IEEE STD 488-1975, Digital Interface for Programmable Instrumentation, with permission of the Institute of Electrical and Electronics Engineers, Inc. Copies of IEEE STD 488-1975 may be purchased from IEEE, 345 East 47th Street, New York, N.Y. 10017, or from the American National Standards Institute, 1430 Broadway, New York, N.Y. 10018.

TABLE OF CONTENTS

	Page
I. Bus Descriptions and Definitions	**1**
A. Bus Line Callouts	1
B. Definitions	1
C. Conventions	4
D. Logic Levels	6
E. Timing	7
F. Connector	8
G. Partitioning	9
II. The Handshake	**12**
A. General Description	12
B. Timing	14
III. State Diagrams	**14**
A. Descriptions and Definitions	14
B. Local Messages	15
C. Remote Messages	15
D. States	15
IV. The MC68488 Interface Adapter	**21**
A. Description and Explanation of Pinouts	21
B. Internal Controls and Registers	22
Data In	23
Data Out	23
Interrupt Mask	23
Interrupt Status	23
Serial Poll	23
Parallel Poll	23
Address Mode	24
Address Status	24
Address Switch	24
Address	24
Auxiliary Command	24
Command Status	26
Command Pass-Through	26
C. Addressing	26
D. Programming Example	28
V. MC3448A Bus Transceiver	**29**
A. Description	29
B. Features and Salient Specifications	29
Conclusion	**33**

Getting Aboard the 488-1975 Bus

I. BUS DESCRIPTIONS AND DEFINITIONS

The IEEE Standard 488-1975 is a byte-serial, bit-parallel interface system primarily defined for programmable measurement instrument systems. The general system defines all circuits, cables, connectors, control protocol and message repertoire to ensure unambiguous data transfer between devices.

The system has the following defined constraints:

1. No more than 15 devices can be interconnected by a single contiguous bus
2. Total transmission length is not to exceed 20 meters, or 2 meters times the number of devices, whichever is less
3. Data rate through any signal line must be less than, or equal to, 1 Megabit/second
4. All data exchange is to be digital (as opposed to analog)

The 488-1975 system is structured with 16 transmission lines. They consist of:

1. 8 data bus lines, permitting transmission of ASCII characters. Data is asynchronous and generally bidirectional.
2. 3 data byte transfer control lines (handshake)
3. 5 general interface management signal lines

These lines may employ either open-collector (not open-collector in an absolute sense due to the termination networks on the bus) or three-state driver elements with certain constraints on the SRQ, NRFD, DIO1–8 and NDAC lines.

A. BUS LINE CALLOUTS

The following are the 16 lines defined in the 488 bus and their definitions:

DIO1-DIO8 Data Input/Output — These are the message lines for carrying data in a bit-parallel, byte-serial form. Data is asynchronous and generally bidirectional. These lines carry either data or address information, depending on the condition of the ATN line.

DAV (Data Valid) — One of the three Handshake lines used to indicate availability and validity of information on the DIO lines. DAV indicates to the acceptor(s) that data is available on the DIO lines.

NRFD (Not Ready for Data) — Another Handshake line used to indicate that all devices are or are not ready to accept data.

NDAC (Not Data Accepted) — The final Handshake line used to indicate the acceptance of data by all devices.

ATN (Attention) — One of five bus management lines used to specify how data on the data lines are to be interpreted and which devices must respond to the data. When ATN is true the DIO1–8 lines carry addresses or commands. When false they carry data. (Controller driven.)

IFC (Interface Clear) — A bus management line which is used to place the interface system in a known quiescent state. All interconnected devices contain some portions of the interface system. IFC puts talkers, listeners into their idle states. (Controller driven.)

SRQ (Service Request) — A bus management line used by a device to indicate a need for service and to request an interrupt of the current events sequence.

REN (Remote Enable) — This management line in conjunction with other messages selects between two alternate sources of device programming data. (Example: front panel control or interface control.) (Controller driven.)

EOI (End Or Identify) — The final management line used to indicate the end of multiple byte transfer sequences or with ATN to perform a parallel polling sequence.

B. DEFINITIONS

A number of definitions of terminology are presented here to aid in understanding subsequent discussions of bus concepts. Unless otherwise noted with an asterisk (*), these definitions are directly from the IEEE Standard.

Active Transfer — A technique for resolving conflicts between two devices simultaneously sending opposite remote message values whereby the interface is so structured that the active value overrides the passive value.

Bidirectional bus — A bus used by any individual device for two-way transmission of messages, that is, both input and output.

Bit-Parallel — Refers to a set of concurrent data bits present on a like number of signal lines used to carry information. Bit-parallel data bits may be acted upon concurrently as a group (byte) or independently as individual data bits.

Bus — A signal line or a set of signal lines used by an interface system to which several devices can be connected and over which messages are carried.

Byte — A group of adjacent binary digits operated on as a unit and usually shorter than a computer word (frequently connotes a group of eight bits).

Byte-Serial — A sequence of bit-parallel data bytes used to carry information over a common bus.

Compatibility — The degree to which devices may be interconnected and used, without modification, when designed as defined throughout Sections 2, 3, and 4 of the 488-1975 Standard.

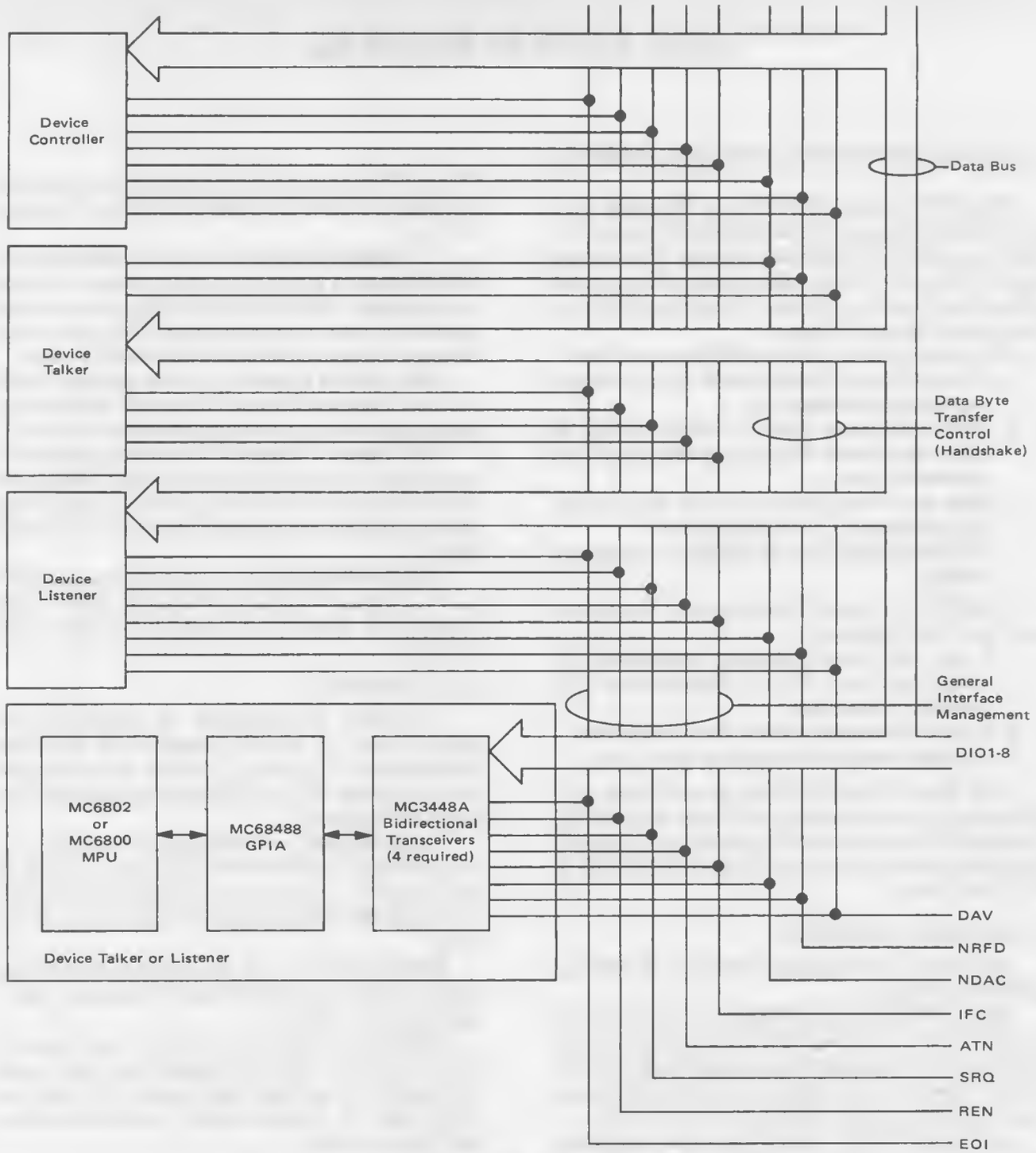


FIGURE 1 — A Possible Bus Structure

Controller — A device that can address other devices to listen or to talk. In addition, this device can send interface messages to command specified actions within other devices. A device with only this capability neither sends nor receives device dependent messages.

NOTE: The use of the word controller throughout the Standard applies strictly to the management (control) of the interface system and does not imply the broad capabilities typically associated with the word in the data processing context. Further classification of the controller is made in Section 2 of IEEE 488-1975 to distinguish between different types of controller capabilities related to the interface system. Listener, talkers, and controller capabilities occur individually and collectively in devices interconnected via the interface system as shown in Figure 1.

Device Dependent Messages — Messages used by the devices interconnected via the interface system that

are carried by, but not used or processed by the interface system directly. Device dependent messages are passed between the device functions and the message coding logic via specified interface functions. These will cause no state transitions within the interface functions. Examples of device dependent messages include device programming data, device measurement data, and device status data (see "Partitioning," Figure 7, message route 3).

An Expression — Consists of one or more local messages, remote messages, state linkages, or minimum time limits used in conjunction with the operators AND, OR, or NOT.

Handshake Cycle — The process whereby digital signals affect the transfer of each data byte across the interface by means of an interlocked sequence of status and control signals. Interlocked denotes a fixed sequence of events in which one event in the sequence must occur before the next event may occur.

High State — The relatively more positive signal level used to assert a specific message content associated with one of two binary logic states.

Interface — A shared boundary between a considered system and another system, or between parts of a system, through which information is conveyed.

Interface Messages — Messages used to manage the interface system itself. Each interface message is sent to cause a state transition within another interface function. An interface message will not be passed along to the device when received by an interface function (see "Partitioning," Figure 7, message route 2).

Interface System — The device-independent mechanical, electrical, and functional elements of an interface necessary to effect communication among a set of devices. Cables, connector, driver and receiver circuits, signal line descriptions, timing and control conventions, and functional logic circuits are typical interface system elements.

Listener — A device that can be addressed by an interface message to receive device dependent messages from another device connected to the interface system.

Listener, talker, and controller capabilities occur individually and collectively in devices interconnected via the interface system as shown in Figure 1.

Local Control — A method whereby a device is programmable by means of its local (front or rear panel) controls in order to enable the device to perform different tasks. (Also referred to as manual control.)

Local Messages — Local messages flow between device functions and interface functions (see "Partitioning," Figure 7, message route 5).

NOTE: Certain local messages are conveyed as remote messages and vice versa.

The designer is not allowed to introduce new local messages to interface functions.

The designer is allowed to introduce a local message derived from any state of any interface functions to device function(s).

Local messages sent by device functions must exist for enough time to cause the required state transitions.

Low State — The relatively less positive signal level used to assert a specific message content associated with one of two binary logic states.

Messages — Quantities of information carried by the interface system. A message will be received either true or false at any specific time.

All communication between an interface function and its environment is accomplished through message sent or received.

Message Coding — Message coding is the act of translating remote messages to or from interface signal lines values.

***Mnemonic** — Refers to a technique of abbreviations that has some easily remembered relationship to the name of the state or message; i.e., NY for New York, JFK for New York's John F. Kennedy Airport.

Multiline Message — A message that shares a group of signal lines with other messages, in some mutually exclusive set. Only one multiline message (message byte) can be sent at one time.

Passive Transfer — A technique for resolving conflicts between two devices simultaneously sending opposite remote message values. A passive value is not guaranteed to be the value received as it can be overridden by an active transfer.

Programmable — That characteristic of a device that makes it capable of accepting data to alter the state of its internal circuitry to perform two or more specific tasks.

Programmable Measuring Apparatus — A measuring apparatus that performs specified operations on command from the system and, if it is a measuring apparatus proper, may transmit the results of the measurement(s) to the system.

Remote Control — A method whereby a device is programmable via its electrical interface connection in order to enable the device to perform different tasks.

Remote Messages — Messages sent via the interface between interface functions of different devices are called remote messages.

Each remote message is either an interface message or a device dependent message.

Signal — The physical representation which conveys data from one point to another.

NOTE: For the purpose of the 488 Standard, this is a restricted definition of what is often called "signal" in more general sense, and is hereinafter referred to digital electrical signals only.

Signal Level — The magnitude of a signal when considered in relation to an arbitrary reference magnitude (voltage in the case of the 488 Standard).

Signal Line — One of a set of signal conductors in an interface system used to transfer messages among interconnected devices.

Signal Parameter — That parameter of an electrical quantity whose values or sequence of values convey information.

State Linkage — The logical interconnection of two interface functions where the transition to an active state of one interface function is dependent on the existence of a specified active state of another interface function (see "Partitioning," Figure 7, message route 4).

System — A set of interconnected elements constituted to achieve a given objective by performing a specified function.

Talker — A device that can be addressed by an interface message to send device dependent messages to another device connected to the interface system.

Listener, talker, and controller capabilities occur individually and collectively in devices interconnected via the interface system as shown in Figure 1.

Terminal Unit — An apparatus by means of which a connection (and translation, if required) is made between the considered interface system and another external interface system.

Unidirectional Bus — A bus used by any individual device for one-way transmission of messages only, that is, either input only or output only.

Uniline Message — A message sent over a single signal line. Two or more of these messages can be sent concurrently.

C. CONVENTIONS

Following the lead of IEEE 488-1975, certain conventions will be followed throughout the remaining text and in all figures and diagrams.

Local Messages — All local messages to an interface function will be represented by three-letter mnemonics in lower case. Example: rdy. Local messages are tabulated in Table 1.

Remote Messages — All remote messages received via the interface will be represented by three-letter mnemonics in upper case. Example: ATN. Remote messages are given in Table 2.

TABLE 1 — Local Message Mnemonics

Mnemonic	Message Title
gts	go to standby
isr	individual service request (qual)
lon	listen only
[lpe]	local poll enable
ltn	listen
lun	local unlisten
nba	new byte available
pon	power on
rdy	ready
rpp	request parallel poll
rsc	request system control
rsv	request service
rtl	return to local
sic	send interface clear
sre	send remote enable
tca	take control asynchronously
tcs	take control synchronously
ton	talk only

TABLE 2 — Remote Message Mnemonics
(See Page 10 for required coding.)

Mnemonic	Message Title
ATN	Attention
DAB	Data Byte
DAC	Data Accepted
DAV	Data Valid
DCL	Device Clear
END	End
GET	Group Execute Trigger
GTL	Go To Local
IDY	Identify
IFC	Interface Clear
LLO	Local Lockout
MLA	My Listen Address
[MLA]	My Listen Address
MSA or [MSA]	My Secondary Address
MTA	My Talk Address
[MTA]	My Talk Address
OSA	Other Secondary Address
OTA	Other Talk Address
PCG	Primary Command Group
PPC	Parallel Poll Configure
[PPD]	Parallel Poll Disable
[PPE]	Parallel Poll Enable
PPRn	Parallel Poll Response n
PPU	Parallel Poll Unconfigure
REN	Remote Enable
RFD	Ready For Data
RQS	Request Service
[SDC]	Selected Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
SQR	Service Request
STB	Status Byte
TCT or [TCT]	Take Control
UNL	Unlisten

State Notation — Each state that an interface function can assume will be represented by a four-letter upper case mnemonic with an S being the final letter. The mnemonic will be encircled in graphical form and all permissible transitions between states of the interface function will be represented by arrows between the circles. States are summarized in Table 3.

Linkage — A linkage from another state diagram will be represented by a four letter mnemonic enclosed in an oval.

Example: (LACS) . Linkages are listed in Table 4.

Maximum Time — If a transition has a maximum time limit, it will be indicated by the symbol t_n. Thus the state pointed to must be entered within a specified amount of time after the expression becomes true. See Table 5.

TABLE 3 — Interface State Mnemonics

Mnemonic	Message Title	Mnemonic	Message Title
ACDS	Accept Data State	PACS	Parallel Poll Addressed To Configure State
ACRS	Acceptor Ready State	PPAS	Parallel Poll Active State
AIDS	Acceptor Idle State	PPIS	Parallel Poll Idle State
ANRS	Acceptor Not Ready State	PPSS	Parallel Poll Standby State
APRS	Affirmative Poll Response State	PUCS	Parallel Poll Unaddressed To Configure State
AWNS	Acceptor Wait For New Cycle State	REMS	Remote State
CACS	Controller Active State	RWLS	Remote With Lockout State
CADS	Controller Addressed State	SACS	System Control Active State
CAWS	Controller Active Wait State	SDYS	Source Delay State
CIDS	Controller Idle State	SGNS	Source Generate State
CPPS	Controller Parallel Poll State	SIAS	System Control Interface Clear Active State
CPWS	Controller Parallel Poll Wait State	SIDS	Source Idle State
CSBS	Controller Standby State	SIIS	System Control Interface Clear Idle State
CSNS	Controller Service Not Requested State	SINS	System Control Interface Clear Not Active State
CSRS	Controller Service Requested State	SIWS	Source Idle Wait State
CSWS	Controller Synchronous Wait State	SNAS	System Control Not Active State
CTRS	Controller Transfer State	SPAS	Serial Poll Active State
DCAS	Device Clear Active State	SPIS	Serial Poll Idle State
DCIS	Device Clear Idle State	SPMS	Serial Poll Mode State
DTAS	Device Trigger Active State	SRAS	System Control Remote Enable Active State
DTIS	Device Trigger Idle State	SRIS	System Control Remote Enable Idle State
LACS	Listener Active State	SRNS	System Control Remote Enable Not Active State
LADS	Listener Addressed State	SRQS	Service Request State
LIDS	Listener Idle State	STRS	Source Transfer State
LOCS	Local State	SWNS	Source Wait For New Cycle State
LPAS	Listener Primary Addressed State	TACS	Talker Active State
LPIS	Listener Primary Idle State	TADS	Talker Addressed State
LWLS	Local With Lockout State	TIDS	Talker Idle State
NPRS	Negative Poll Response State	TPIS	Talker Primary Idle State

Minimum Time — A minimum time limit is represented by the symbol T_n. This symbol achieves a true value only after the interface has been in the state originating the transition for the time value specified. It will remain true until the state is exited. See Table 5 for these time limits.

AND — The AND operator is represented by the symbol $\wedge$.

OR — The OR operator is represented by the symbol $\vee$. (The AND operator takes precedence over the OR operator unless specified by parenthesis.)

NOT — The NOT operator is represented by a horizontal bar above the portion of the expression to be negated.

Active/Passive — The interface is so structured that conflicts which may arise when opposite remote message values are simultaneously transmitted by two devices are resolved. In such cases one message must always be made to override the other. Thus an active value will override a passive value when conflicts occur.

T — indicates active true
F — indicates active false
(T) — indicates passive true
(F) — indicates passive false

Optional True — If a portion of an expression is optional in that its true value is not required for the complete expression to be true, then it will be enclosed within square brackets [.....].

TABLE 4 — State Linkages

State Diagram	Mnemonic	Interface State
SH	TACS	Talker Active State (T function)
SH	SPAS	Serial Poll Active State (T function)
SH	CACS	Controller Active State (C function)
SH	CTRS	Controller Transfer State (C function)
AH	LADS	Listener Addressed State (L function)
AH	LACS	Listener Active State (L function)
T	ACDS	Accept Data State (AH function)
TE	ACDS	Accept Data State (AH function)
TE	LPAS	Listener Primary Addressed State (L function)
L	ACDS	Accept Data State (AH function)
L	CACS	Controller Active State (C function)
LE	ACDS	Accept Data State (AH function)
LE	CACS	Controller Active State (C function)
LE	TAPS	Talker Primary Addressed State (T function)
SR	SPAS	Serial Poll Active State (T function)
RL	ACDS	Accept Data State (AH function)
RL	LADS	Listener Addressed State (L function)
PP	ACDS	Accept Data State (AH function)
PP	LADS	Listener Addressed State (L function)
DC	ACDS	Accept Data State (AH function)
DC	LADS	Listener Addressed State (L function)
DT	ACDS	Accept Data State (AH function)
DT	LADS	Listener Addressed State (L function)
C	ACDS	Accept Data State (AH function)
C	ANRS	Acceptor Not Ready State (AH function)
C	STRS	Source Transfer State (SH function)
C	TADS	Talker Addressed State (T function)

D. LOGIC LEVELS

Throughout the 488-1975 document the coded logical states Ø and 1 are utilized. A negative logic convention is defined as follows:

Coding Logical State	Electrical Signal Level
Ø = False	≥2.Ø V Called the high state.
1 = True	≤Ø.8 V Called the low state.

Either open-collector or three-state drivers may be utilized with the following constraints:

1. Open collector types are required on the following lines: SRQ, NRFD, and NDAC. In addition, if parallel polling is used DIO1–DIO8 must also be open collector types.
2. Three-state drivers are recommended for higher speed systems and especially in a controller ATN signal line if used with other devices employing three-state drivers on the DAV, EOI and DIO1–DIO8 lines.

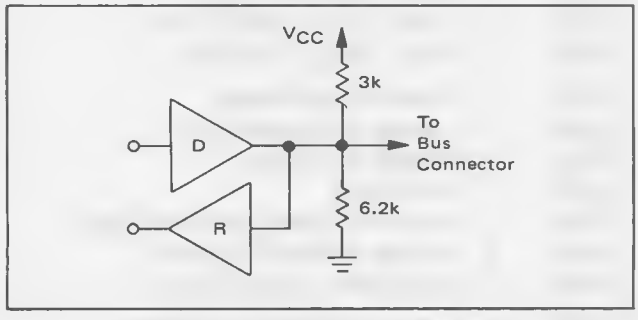


FIGURE 2 — Typical TTL Implementation

Drivers of either open-collector or three-state configuration must be capable of sinking +48 mA without exceeding 0.4 V. (At the time of printing consideration was being given extending the limit to 0.5 V to permit devices employing Schottky technology.) Three-state drivers must maintain 2.4 V or greater when sourcing 5.2 mA. Typical specification for both open-collector and three-state drivers and receivers are given in the 488 standard. When implemented with standard TTL logic elements and with a 3 kΩ resistor to V_{CC} and a 6.2 kΩ resistor to ground at each common node, the typical suggested configuration is met (see Figure 2). The actual dc load requirement which must be met, however, is a function of driver, receiver and resistive terminations. This load requirement is given in Figure 3.

The receivers may be standard TTL gates with 2.0 V high state and 0.8 V low state input voltages or Schmitt trigger types employing hysteresis.

If hysteresis is utilized for improved noise immunity, it is recommended that at least 0.4 V be employed. It is also recommended that the lower threshold be greater than 0.8 V and the upper threshold less than +2.0 V. Negative voltage clamping is also required within the receiver.

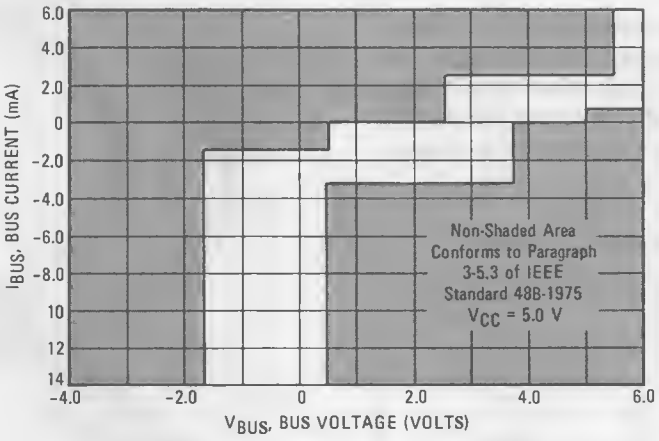


FIGURE 3 — Typical Bus Load Line

E. TIMING

All specified timing constraints are summarized in Table 5.

TABLE 5 — Time Values

Time Value Identifier*	Function (applies to)	Description	Value
T_1	SH	Settling time for multiline messages	≥2 μs†
t_2	SH,AH,T,L	Response to ATN	≤200 ns
T_3	AH	Interface message accept time‡	>0§
t_4	T,TE,L,LE,C	Response to IFC or REN false	<100 μs
t_5	PP	Response to ATN ∧ EOI	≤200 ns
T_6	C	Parallel poll execution time	≥2 μs
T_7	C	Controller delay to allow current talker to see ATN message	≥500 ns
T_8	C	Length of IFC or REN false	>100 μs
T_9	C	Delay for EOI**	≥1.5 μs††

*Time values specified by a lower case t indicate the maximum time allowed to make a state transition. Time values specified by an upper case T indicate the minimum time that a function must remain in a state before exiting.

†If three-state drivers are used on the DIO, DAV, and EOI lines, T_1 may be:
(1) ≥1100 ns
(2) Or ≥700 ns if it is known that within the controller ATN is driven by a three-state driver
(3) Or ≥500 ns for all subsequent bytes following the first sent after each false transition of ATN (the first byte must be sent in accordance with (1) or (2))

‡Time required for interface functions to accept, not necessarily respond to interface messages.

§Implementation dependent.

**Delay required for EOI, NDAC, and NRFD signal lines to indicate valid states.

†† 600 ns for three-state drivers.

F. THE CONNECTOR

Both the dimensions and the actual pin locations of the connector are prescribed in the 488-1975 Standard. Recommended connectors include MICRORIBBON (Amphenol or Cinch Series 57) or CHAMP (AMP). Illustrations of the connector, cables and pin connections are presented in Figures 4 thru 6.

TABLE 6 — Connector Pin Assignments

Contact	Signal Line	Contact	Signal Line
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	EOI	17	REN
6	DAV	18	Gnd, (6)
7	NRFD	19	Gnd, (7)
8	NDAC	20	Gnd, (8)
9	IFC	21	Gnd, (9)
10	SRQ	22	Gnd, (10)
11	ATN	23	Gnd, (11)
12	SHIELD	24	Gnd, LOGIC

NOTE: Gnd, *(n)* refers to the signal ground return of the referenced contact.

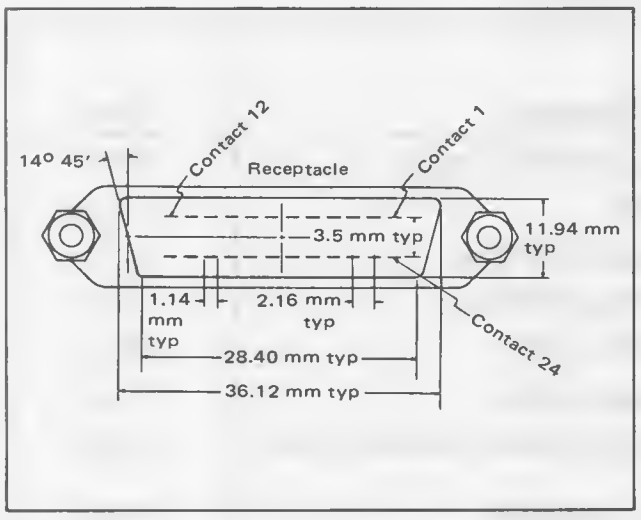


FIGURE 4 — Device Connector Mounting

Interconnecting cables used with interface systems have dual connectors. These can be stacked to accommodate a variety of physical layouts by allowing more than one cable to be attached to any device.

FIGURE 5 — Cable Connectors

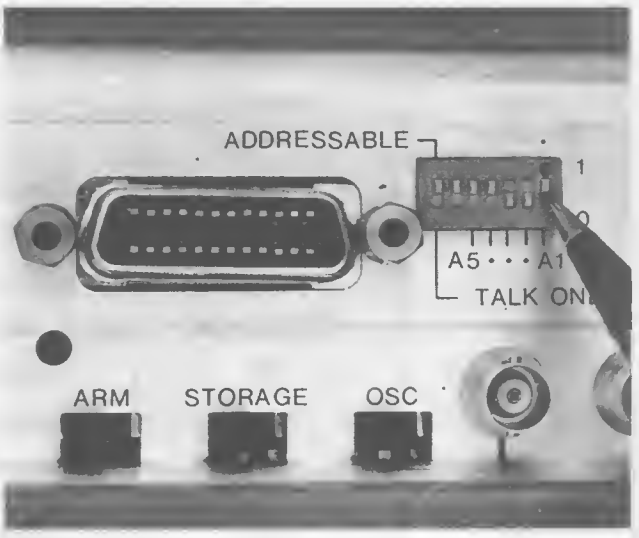


FIGURE 6 — Device-Mounted Connector

Photos courtesy of Hewlett-Packard.

G. PARTITIONING

The 488-1975 system can be divided into three major functional parts: (1) the device function, (2) the interface functions, and (3) the message coding logic.

The device function is simply the application the device has been designed to perform, i.e. voltmeter, signal generator, logic analyzer, etc. The interface function is that part of the system which allows for the basic link through which a device can receive, process, and transmit messages.

Message coding is the process of converting remote messages to or from interface signal line values.

There are two types of messages: (1) uniline, a message sent over a single line, and (2) multiline, a message that uses a group of lines. Only one multiline message may be sent at a time. However, more than one uniline message may be sent at a time (on different lines).

The allowable Remote Messages and the required coding to send or receive these messages are shown in Table 7. Lines not specified in Table 5 must not be driven and must be ignored by the receiver.

A uniline message value is valid as soon as the corresponding logic state is detected. However, multiline messages are valid only within the context of SH and AH functions. Thus the transmitted multiline message is valid while the SH function is in the STRS state. The received multiline message is valid while the AH function is in the ACDS state. All passive message values are transferred as Ø signal line states. This requires the logic OR of signal line states to be performed on the interface.

Partitioning is summarized in Figure 7.

The drivers and receivers are necessary to provide the necessary current levels and termination networks to satisfy the requirements in the preceeding section on logic levels. It is generally assumed that either bipolar or MOS technology can be employed for the message coding and interface function sections. MOS LSI will probably be preferred due to the logic complexity and either gold-doped or Schottky bipolar devices can be utilized to implement the 48 mA drivers to provide the current drive at the actual bus.

The proposed Motorola complement of IC's uses a Schottky bipolar quad transceiver, type MC3448A, for the driver/receiver function, and an NMOS LSI device, type MC68488, to provide the message coding and interface functions. The MC68488 is designed specifically to be an interface between the 488-1975 bus and the M6800 microprocessor bus. With some modification, however, the MC68488 can be made to function with other microprocessor systems.

Details of these two IC's will be presented in Sections IV and V.

A number of basic interface functions are summarized in Tables 8 and 9.

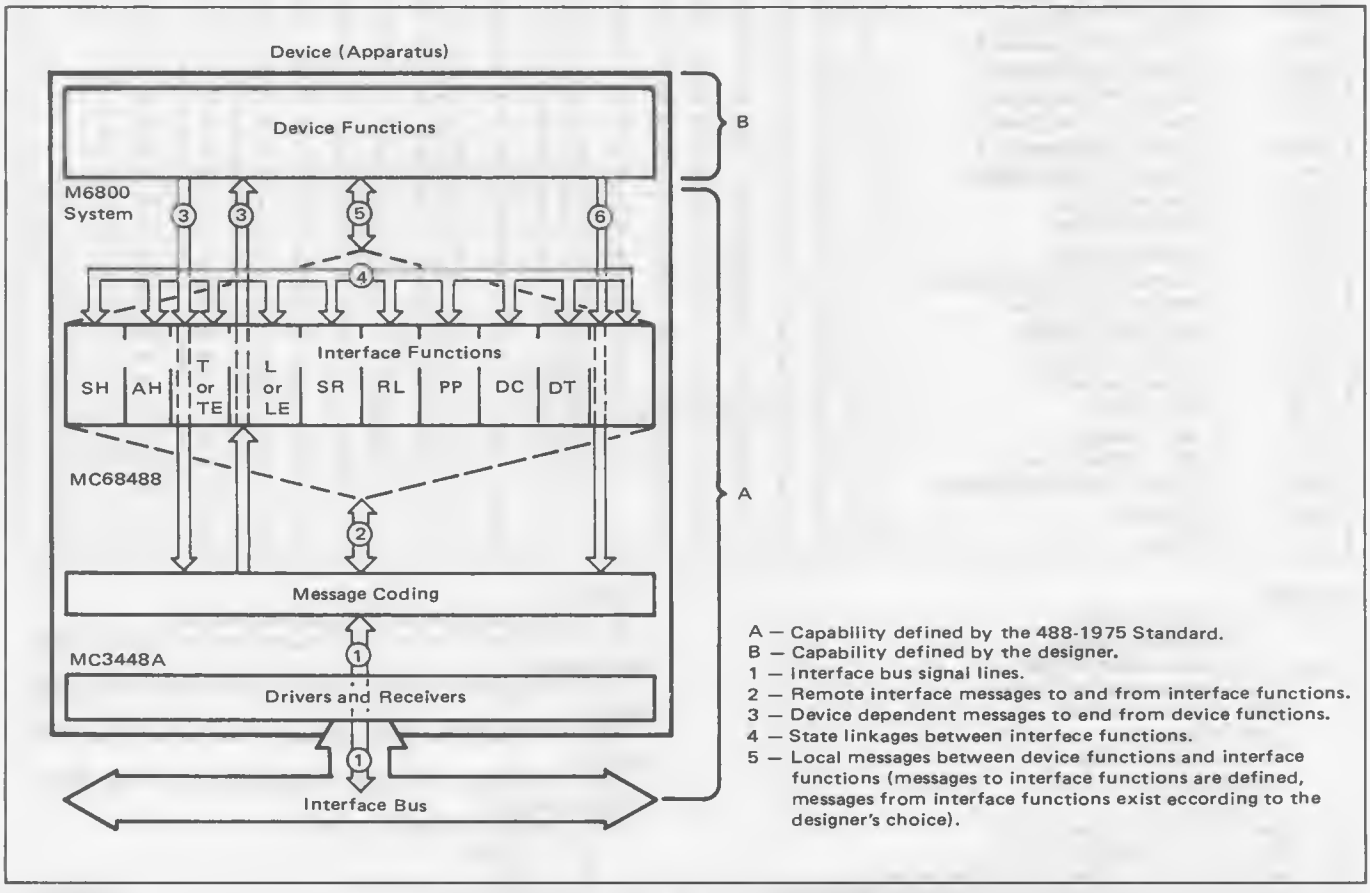


FIGURE 7 — Functional Partition Within a Device

TABLE 7 — Remote Message Coding

Mnemonic	Message Name	Bus Signal Line(s) and Coding That Assert Message True															Message Type	Message Class	Notes
		Data I/O								Handshake			Bus Management						
		DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DAV	NRFD	NDAC	ATN	EOI	RSQ	IFC	REN			
ACG	Addressed Command Group	X	Ø	Ø	Ø	X	X	X	X	X	X	X	X	X	X	X	X	M	AC	10
ATN	Attention	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	U	UC	—
DAB	Data Byte	D8	D7	D6	D5	D4	D3	D2	D1	X	X	X	X	X	X	X	X	M	DD	1, 9
DAC	Data Accepted	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	U	HS	—
DCL	Device Clear	X	Ø	Ø	1	Ø	1	Ø	Ø	X	X	X	X	X	X	X	X	M	UC	10
END	End	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	U	ST	9, 11
EOS	End of String	E8	E7	E6	E5	E4	E3	E2	E1	X	X	X	X	X	X	X	X	M	DD	2, 9
GET	Group Execute Trigger	X	Ø	Ø	Ø	1	Ø	Ø	Ø	X	X	X	X	X	X	X	X	M	AC	10
GEL	Go To Local	X	Ø	Ø	Ø	Ø	Ø	Ø	1	X	X	X	X	X	X	X	X	M	AC	10
IDY	Identify	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	U	UC	10, 11
IFC	Interface Clear	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	U	UC	—
LAG	Listen Address Group	X	Ø	1	X	X	X	X	X	X	X	X	X	X	X	X	X	M	AD	10
LLO	Local Lock Out	X	Ø	Ø	1	Ø	Ø	Ø	1	X	X	X	X	X	X	X	X	M	UC	10
MLA	My Listen Address	X	Ø	1	L5	L4	L3	L2	L1	X	X	X	X	X	X	X	X	M	AD	3, 10
MTA	My Talk Address	X	1	Ø	T5	T4	T3	T2	T1	X	X	X	X	X	X	X	X	M	AD	4, 10
MSA	My Secondary Address	X	1	1	S5	S4	S3	S2	S1	X	X	X	X	X	X	X	X	M	SE	5, 10
NUL	Null Byte	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	X	X	X	X	X	X	X	X	M	DD	—
OSA	Other Secondary Address	(OSA = SCG ∧ $\overline{\text{MSA}}$)																M	SE	10
OTA	Other Talk Address	(OTA = TAG ∧ $\overline{\text{MTA}}$)																M	AD	10
PCG	Primary Command Group	(PCG = ACG ∨ UCG ∨ LAG ∨ TAG)																M	—	10
PPC	Parallel Poll Configure	X	Ø	Ø	Ø	Ø	1	Ø	1	X	X	X	X	X	X	X	X	M	AC	10
PPE	Parallel Poll Enable	X	1	1	Ø	S	P3	P2	P1	X	X	X	X	X	X	X	X	M	SE	6, 10
PPD	Parallel Poll Disable	X	1	1	1	D4	D3	D2	D1	X	X	X	X	X	X	X	X	M	SE	7, 10
PPR1	Parallel Poll Response 1	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	U	ST	—
PPR2	Parallel Poll Response 2	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	U	ST	—
PPR3	Parallel Poll Response 3	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	U	ST	—
PPR4	Parallel Poll Response 4	X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	U	ST	—
PPR5	Parallel Poll Response 5	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	—
PPR6	Parallel Poll Response 6	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	—
PPR7	Parallel Poll Response 7	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	—
PPR8	Parallel Poll Response 8	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	—
PPU	Parallel Poll Unconfigure	X	Ø	Ø	1	Ø	1	Ø	1	X	X	X	X	X	X	X	X	M	UC	10
REN	Remote Enable	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	U	UC	—
RFD	Ready For Data	X	X	X	X	X	X	X	X	X	Ø	X	X	X	X	X	X	U	HS	—
RQS	Request Service	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	9
SCG	Secondary Command Group	X	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	M	SE	10
SDC	Selected Device Clear	X	Ø	Ø	Ø	Ø	1	Ø	Ø	X	X	X	X	X	X	X	X	M	AC	10
SPD	Serial Poll Disable	X	Ø	Ø	1	1	Ø	Ø	1	X	X	X	X	X	X	X	X	M	UC	10
SPE	Serial Poll Enable	X	Ø	Ø	1	1	Ø	Ø	Ø	X	X	X	X	X	X	X	X	M	UC	10
SRQ	Service Request	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	U	ST	—
STB	Status Byte	S8	X	S6	S5	S4	S3	S2	S1	X	X	X	X	X	X	X	X	M	ST	8, 9
TCT	Take Control	X	Ø	Ø	Ø	1	Ø	Ø	1	X	X	X	X	X	X	X	X	M	AC	10
TAG	Talk Address Group	X	1	Ø	X	X	X	X	X	X	X	X	X	X	X	X	X	M	AD	10
UGG	Universal Command Group	X	Ø	Ø	1	X	X	X	X	X	X	X	X	X	X	X	X	M	UC	10
UNL	Unlisten	X	Ø	1	1	1	1	1	1	X	X	X	X	X	X	X	X	M	AD	10
UNT	Untalk	X	1	Ø	1	1	1	1	1	X	X	X	X	X	X	X	X	M	AD	10

Symbols:
Type U = Uniline message
M = Multiline message
Class AC = Addressed command
AD = Address (talk or listen)
DD = Device dependent
HS = Handshake
UC = Universal Command
SE = Secondary
ST = Status
Ø = logical zero (HIGH Signal Level)
1 = logical one (LOW Signal Level)
X = don't care (for the coding of a received message)
X = must not drive (for the coding of a transmitted message)

NOTES:
(1) D1–D8 specify the device dependent data bits.
(2) E1–E8 specify the device dependent code used to indicate the EOS message.
(3) L1–L5 specify the device dependent bits of the device's listen address.
(4) T1–T5 specify the device dependent bits of the device's talk address.
(5) S1–S5 specify the device dependent bits of the device's secondary address.
(6) S specifies the sense of the PPR.

S	Response
Ø	Ø
1	1

(continued)

P1–P3 specify the PPR message to be sent when a parallel poll is executed.

P3	P2	P1	PPR Message
ø	ø	ø	PPR1
.	.	.	.
.	.	.	.
.	.	.	.
1	1	1	PPR8

(7) D1–D4 specify don't-care bits that must be sent all zeroes, but do not need to be decoded by the receiving device.

(8) S1–S6, S8 specify the device dependent status. (DIO7 is used for the RQS message.)

(9) The true message value must be ignored when received if the LACS is inactive.

(10) The true message value must be ignored when received if the ATN message is false.

(11) Interface protocol specifies that the IDY message is sent true only when the ATN message is sent true, whereas the END message is sent true only when the ATN message is sent false.

TABLE 8—Basic Interface Functions

Function	Description
TALKER	
Basic Talker	To let an instrument send data to another instrument.
Talk Only	To let an instrument operate in a system without a controller.
Unaddress if my listen address (MLA)	To prevent an instrument capable of functioning as both a talker and a listener from talking to itself.
Extended Talker (TE)	Same as talker function with added addressing capability.
Serial Poll	To send a "status byte" to the controller and identify itself as the source of a service request.
LISTENER	
Basic Listener	To let an instrument receive data from another instrument.
Listen Only	To let an instrument operate in a system without a controller.
Unaddress if my talk address (MTA)	To prevent an instrument capable of functioning as both talker and listener from listening to itself.
Extended Listener (LE)	Same as listener function with added addressing capability.
SOURCE HANDSHAKE	To synchronize the transmission of information on the data bus by the talker when sending instrument-generated data and by the controller when sending interface messages.
ACCEPTOR HANDSHAKE	To synchronize the receipt of information on the data bus for all interface functions when receiving interface messages and for the listener function when receiving instrument-generated data.
CONTROLLER	
System Controller	To let an instrument send the interface clear (IFC) or remote enable (REN) messages.
Send Interface Clear (IFC)	To let a system controller take charge from another controller and/or initialize the bus.
Send Remote Enable (REN)	To let a system controller enable instruments to switch to remote control.
Respond to Service Requests (SRQ)	To let a controller respond to service requests.
Send Interface Messages	To let the controller send multiline interface messages.
Receive Control	To let the controller accept control on the bus from another controller.
Pass Control	To let the controller pass control of the bus to another controller.
Parallel Poll	To let the control execute a parallel poll.
Take Control Synchronously	To let the controller take control of the bus without destroying a data transmission in progress.

TABLE 9—Supplementary Interface Functions

Function	Description
SERVICE REQUEST	To let an instrument indicate to the controller that some event has occurred and request it to take some specific action asynchronously with respect to other bus operations (one SRQ function is required for each independent reason for requesting service).
REMOTE-LOCAL	
Basic Remote-Local	To let the control of the instrument be switched between its local (manual) controls and remote control (programming codes received while addressed as a listener).
Local Lock Out	To let the local control "return to local" be disabled.
PARALLEL POLL	
Basic Parallel Poll	To let instruments return one-bit status to the controller. Up to eight instruments may respond simultaneously. More than one instrument may respond on the same status line so that logical operations (AND, OR) may be performed on a group of instruments.
Parallel Poll Configure	To let the instrument be configured by the controller.
DEVICE CLEAR	
Basic Device Clear	To provide a means by which an instrument (device) may be initialized to a predefined state. All instruments are cleared concurrently.
Selective Device Clear	To clear individual instruments (devices) selectively.
DEVICE TRIGGER	To let instruments (devices), either singly or in a group, be triggered, or some action be started.

II THE HANDSHAKE*

A. GENERAL DESCRIPTION

When data is transferred from a source to one or more acceptors, a handshaking procedure is utilized to ensure that the transfer is performed properly. The procedure is repeated for each byte transferred. Three signal lines are utilized for the handshake: the DAV, NRFD and NDAC lines. The DAV line is controlled by the talker, while the NRFD and NDAC are under the control of the listeners.

The handshake procedure ensures that each listener is ready to accept data, that the data on the DIO1–DIO8 lines is valid data and that the data has been accepted by all listeners. Data will be sent only as rapidly as it can be accepted by the slowest listener.

The three-wire handshake has three important characteristics which give the interface system wide flexibility. First, the data transfer is asynchronous, thus avoiding inherent timing restrictions. Data can be transferred at any rate up to 1 megabyte per second that is suitable to the devices on the bus.

Second, the handshake allows the interconnection of devices which operate at different input/output speeds. Data is transferred automatically at the speed which can be handled by the slowest active device on the bus.

Third, more than one device can accept data simultaneously.

A handshake cycle will be illustrated in a step-by-step manner, with the aid of Figure 8:

Both the source and acceptor power-up or start in a known quiescent state; thus the source has DAV high indicating the data on the bus is not valid, and the acceptor sets both NRFD and NDAC passive true (low) indicating that data has neither been accepted nor is the listener ready to accept data. If both NRFD and NDAC are sensed high by the source, an error condition exists.

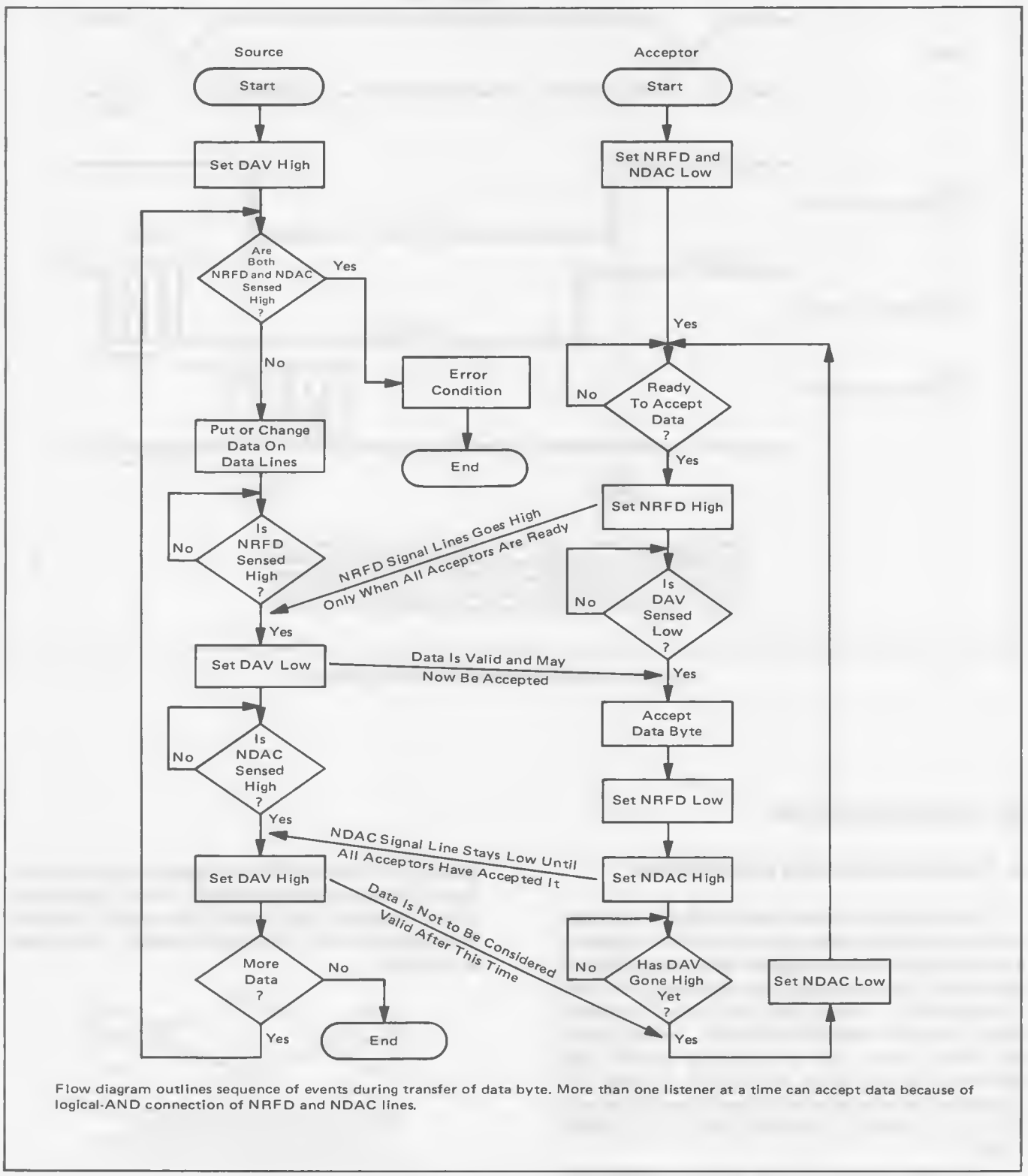
A source will indicate that valid data is available on the DIO lines by setting DAV low. When this event occurs the acceptor(s) will respond by pulling NRFD low. During this time data transfer takes place. Since each acceptor will accept data at different rates their NDAC handshake lines will be set high accordingly. The source will not see NDAC high until the slowest

---

***NOTE:** The three-wire handshake described in this section is the subject of patents owned by Hewlett-Packard Co. Inquiries on license details should be directed to the legal department of Hewlett-Packard.

acceptor has responded. At this time NDAC will go high, indicating that all the acceptors have accepted the data. When the source senses NDAC high the source will set DAV high, indicating that data on the DIO lines is no longer valid. Upon DAV going high, the acceptors will return the NDAC line low. As data transfer continues the cycle repeats.

No step in this sequence can be initiated until the previous step is completed. Thus, information can proceed as fast as devices can respond, but no faster than the slowest device that is presently addressed as active.



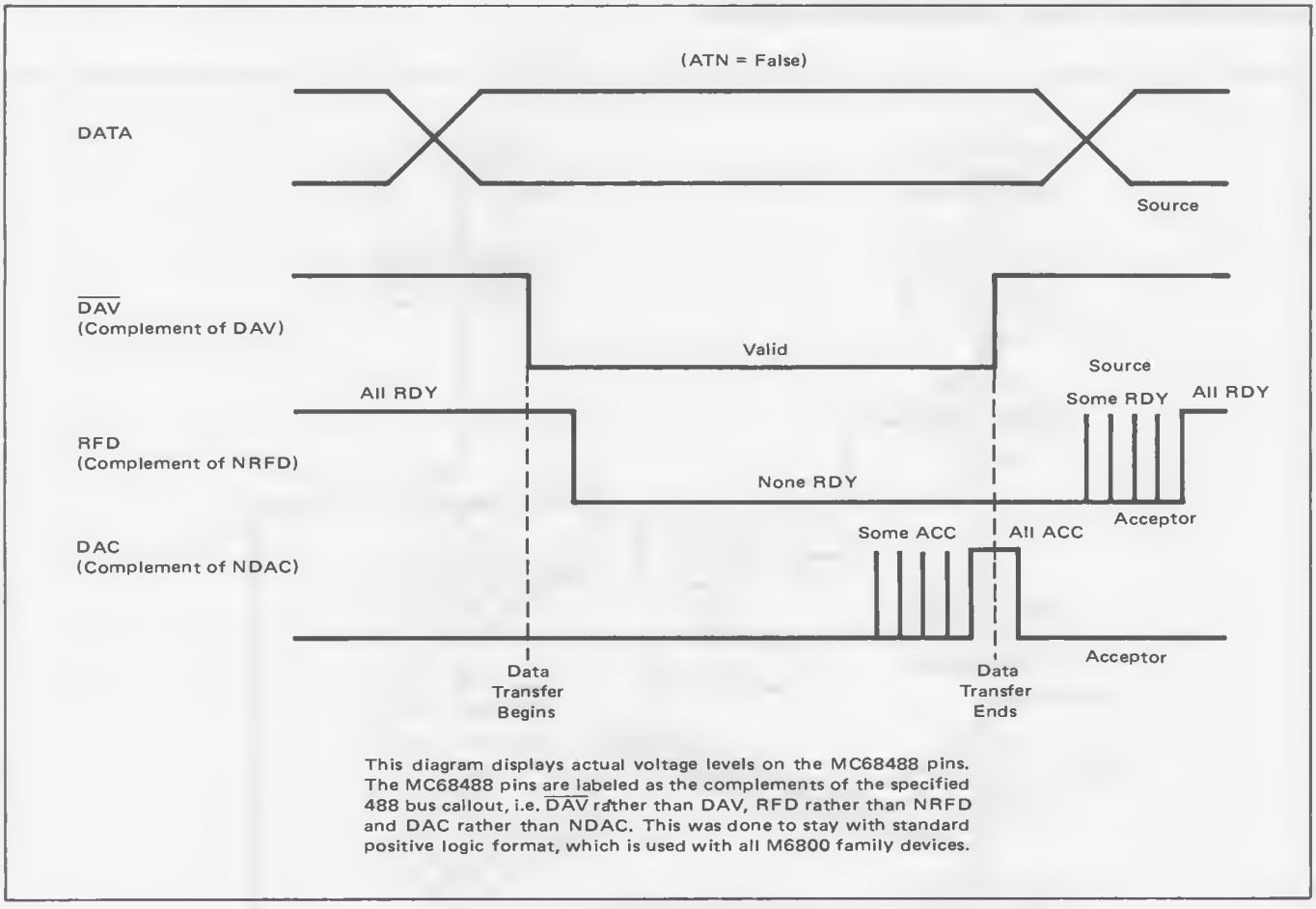
Flow diagram outlines sequence of events during transfer of data byte. More than one listener at a time can accept data because of logical-AND connection of NRFD and NDAC lines.

FIGURE 8 – Data Transfer

B. TIMING OF HANDSHAKE

The sequences of timing for a handshake and byte transfer are illustrated in Figure 9.

Details of the Source Handshake (SH) and Acceptor Handshake (AH) states are given in the State Diagram Section to follow.



This diagram displays actual voltage levels on the MC68488 pins. The MC68488 pins are labeled as the complements of the specified 488 bus callout, i.e. $\overline{DAV}$ rather than DAV, RFD rather than NRFD and DAC rather than NDAC. This was done to stay with standard positive logic format, which is used with all M6800 family devices.

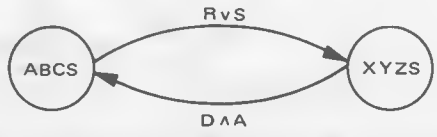
FIGURE 9 – Source and Acceptor Handshake Function

III. STATE DIAGRAMS

A. DESCRIPTIONS AND DEFINITIONS

State diagrams are used throughout the 488 Standard to graphically illustrate the permissible transitions between states and the conditions required to effect such transitions. In the state diagram convention, each state is illustrated as a four-letter upper case mnemonic ending with an S, enclosed with a circle. Arrows connect the various states with the direction of the arrow indicating the path from one state to the next. Expressions beside the path define conditions which must be true to result in the transition along the path of the arrow.

A simple example will be given for illustration. ABCS and XYZS are States. To cause a transition from State ABCS to State XYZS the logical OR of conditions R and S must be true (i.e. either R or S or both must be true). A transition back from XYZS to ABCS will occur when the logical AND of D and A is true (i.e. both D and A are true).



A transition from ABCS to XYZS will occur if and only if RvS (logical OR) occurs.
A transition from XYZS to ABCS will occur if and only if D∧A (logical AND) occurs.

FIGURE 10 – State Diagram

The expressions A, D, R and S could be local messages, remote messages, State linkages, and time limits used in conjunction with the AND, OR or NOT operators.

B. LOCAL MESSAGES

Table 1 gives the mnemonics for the allowable local messages. These messages are sent between a device function and an interface function and are implemented by front panel controls on the device. Local messages carry three-letter, lower case mnemonic labels.

C. REMOTE MESSAGES

Messages passed along the bus between interface functions of different devices are called Remote Messages. Table 2 shows a listing of these three-letter, upper case mnemonics. Table 7 gives the bus signal coding for the remote messages.

D. STATES

State diagram will be given in this section for each of the major functions. The reader should consult Tables 1, 2, 3 and 4 for mnemonics used for local and remote messages, States and linkages respectively in the following State diagrams.

Source Handshake (SH) is the interface function which controls the starting and stopping of multiline message byte transfer.

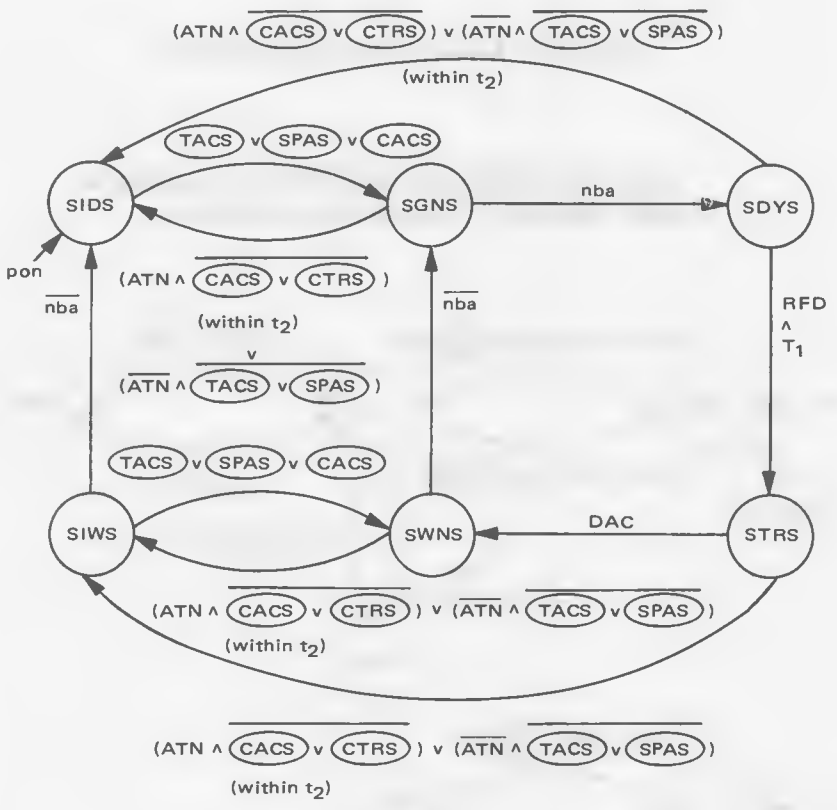
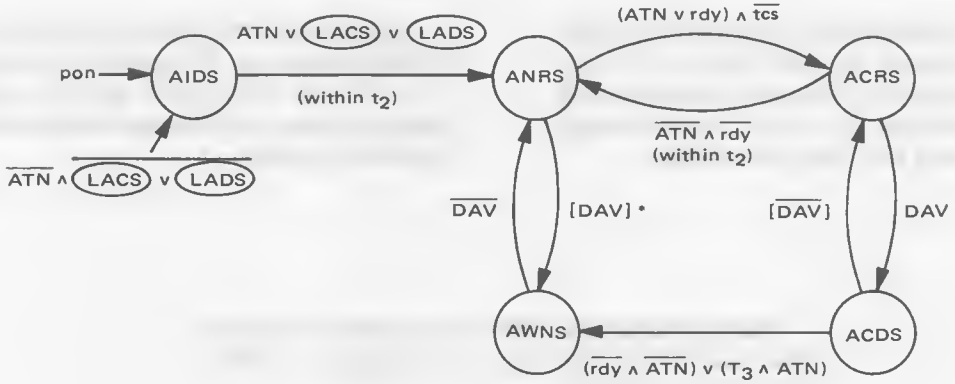


FIGURE 11 — Source Handshake State Diagram

Acceptor Handshake (AH) allows for the proper reception of remote multiline messages within a device. The Acceptor Handshake function uses the DAV, NRFD, and NDAC lines to coordinate each message byte transfer.



*This transition will never occur under normal interface operation; however, it may be implemented to simplify the interface function design.

FIGURE 12 – Acceptor Handshake State Diagram

Talker (T)/Talker Extended (TE) allow the device to transmit data over the interface to other devices.

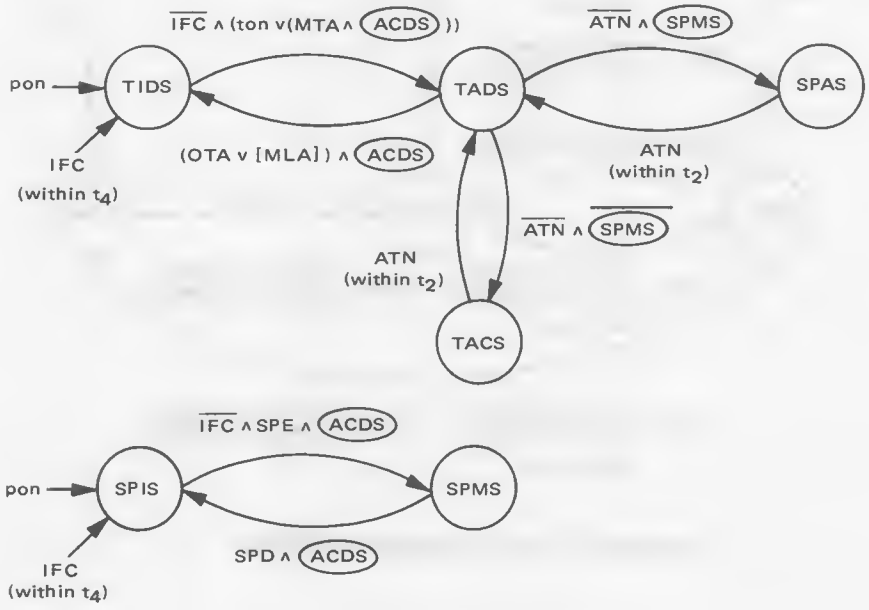


FIGURE 13 – Talker State Diagram

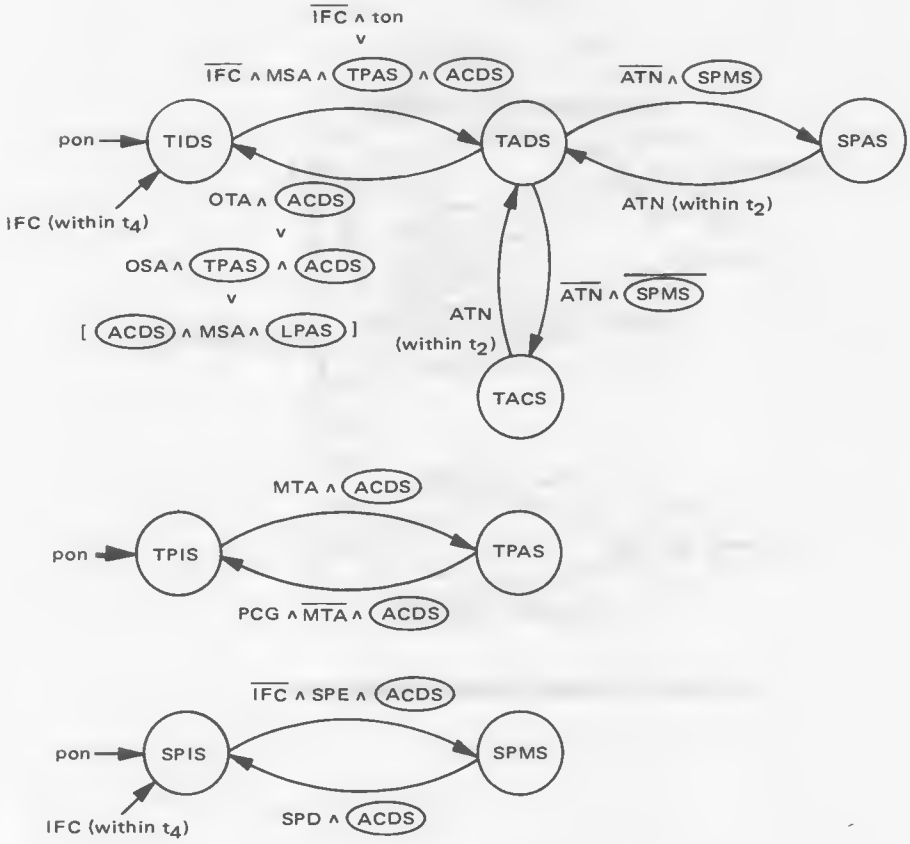


FIGURE 14 — Talker Extended Diagram

Listener (L)/Listener Extended (LE) provide the device with the ability to receive data over the interface bus from other devices.

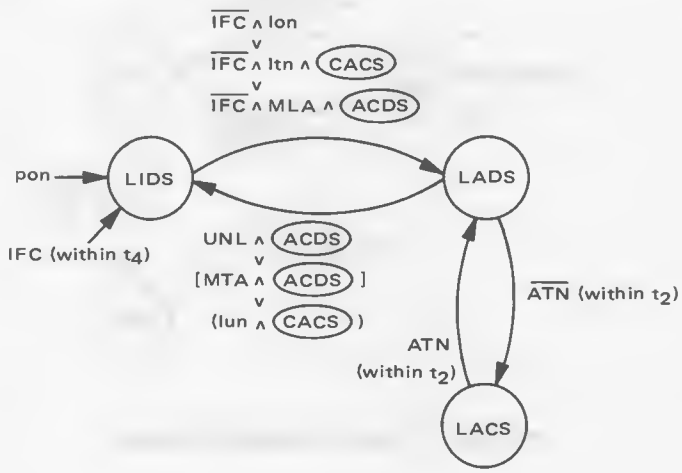


FIGURE 15 — Listener State Diagram

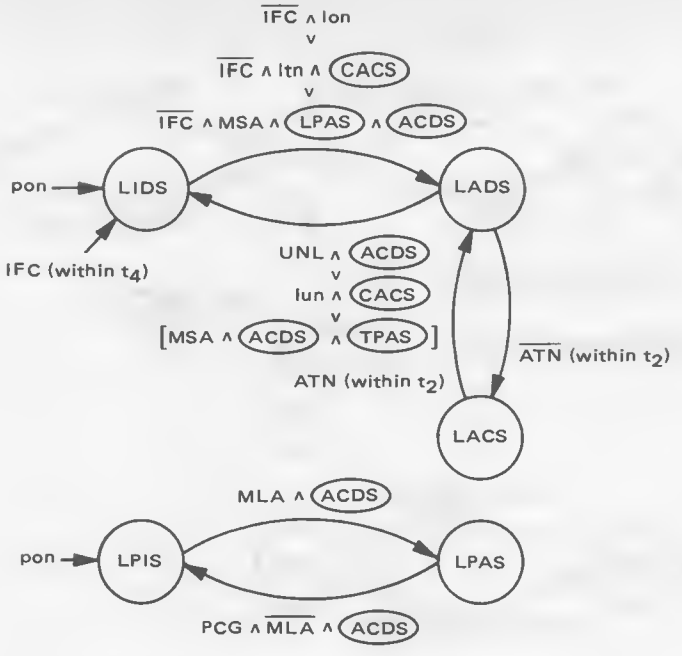


FIGURE 16 – Listener Extended State Diagram

Service Request (SRQ) allows a device to request service from the interface controller.

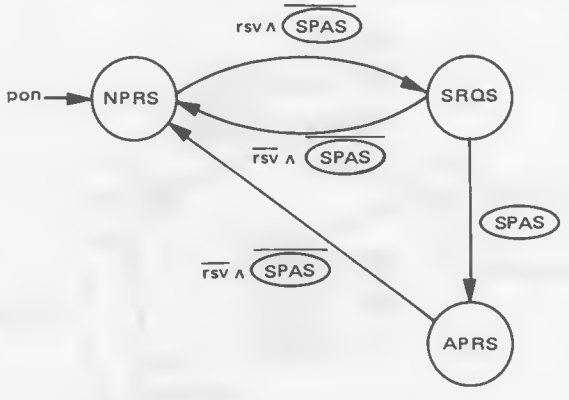


FIGURE 17 – Service Request State Diagram

Remote Local (RL) Allows a device to select from one of two sources of information. The source can be local (input information from the device's front panel) or remote (input information from the interface).

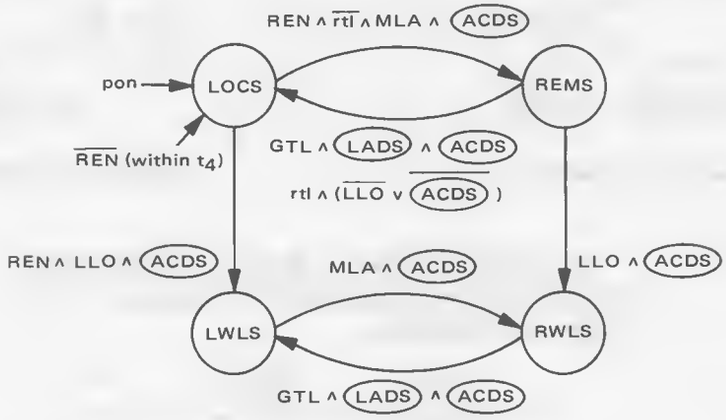


FIGURE 18 – Remote Local State Diagram

Parallel Poll (PP) allows a device to send one bit of status to the controller without having been previously addressed to be a talker.

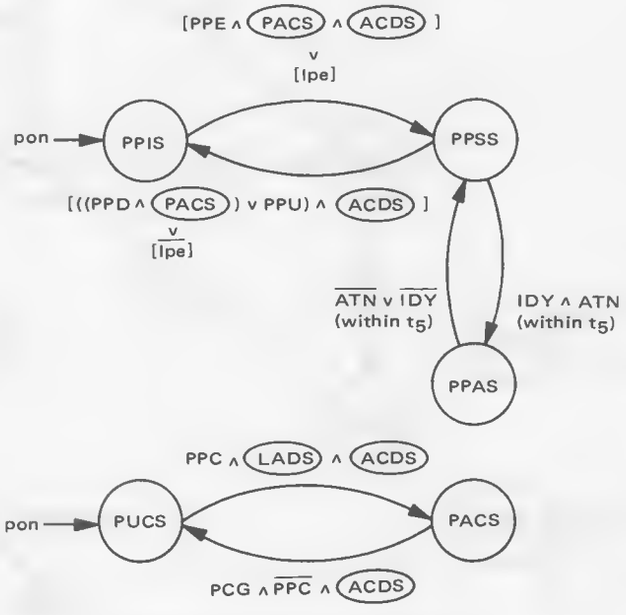


FIGURE 19 – Parallel Poll State Diagram

Device Clear (DC) allows a device to be cleared or initialized alone or as part of a group.

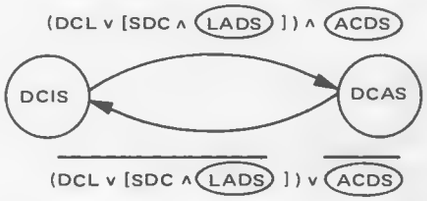


FIGURE 20 — Device Clear State Diagram

Device Trigger (DT) allows a device to be started alone or as part of a group.

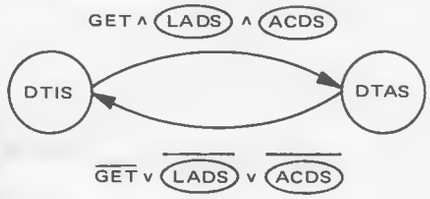


FIGURE 21 — Device Trigger State Diagram

Controller (C) allows a device to send addresses, universal commands, and addressed commands to other devices on the interface.

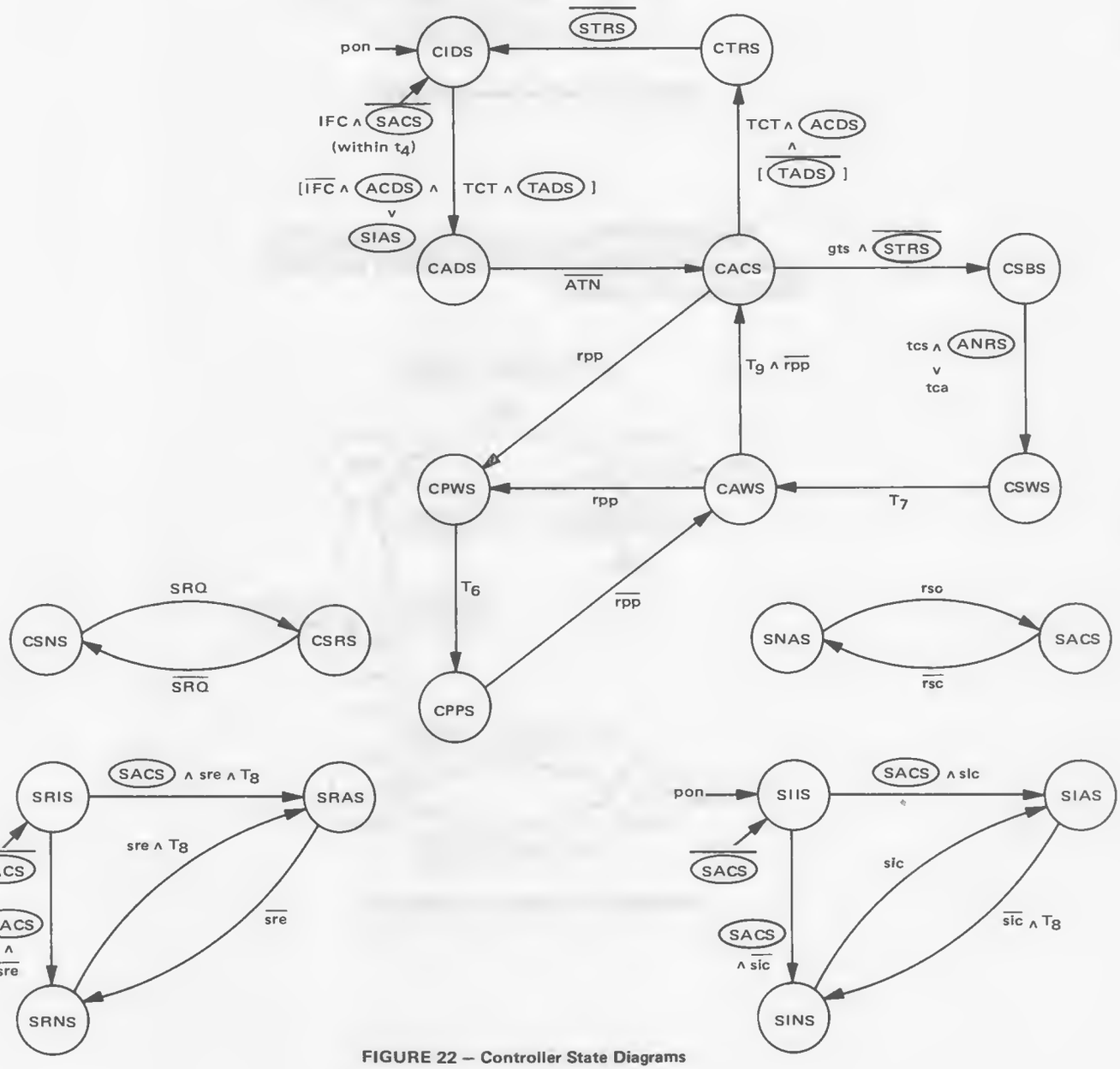


FIGURE 22 — Controller State Diagrams

IV. MC68488 GENERAL PURPOSE INTERFACE ADAPTER (GPIA)

A. DESCRIPTION AND EXPLANATION OF PINOUTS

This new LSI NMOS device provides the interface function between the IEEE 488 Standard instrument bus and the M6800 family MPU bus. The MC6800 can receive, process, and send messages to the interface system through the MC68488. The MC68488 does not make the MC6800 a stand-alone controller. However, the MC6800/MC68488 can implement the device controller function with some additional external logic and applicable software.

The MC68488 is able to automatically handle the following interface protocol.

- Single or dual address capability
- Secondary address capability
- Complete source and acceptor handshake
- Complete talker and listener state diagrams
- Service request state diagram
- Local lockout
- Parallel poll state diagram
- Device clear state diagram
- Device trigger state diagram
- Ready for data (RFD) hold off

In addition, the MC68488 also has the additional capabilities related to the M6800 family.

- Programmable Interrupts
- Trigger output pins
- DMA (Direct Memory Access) lines
- Address Switch Enable ($\overline{ASE}$) output
- Two transmit/$\overline{\text{receive}}$ buffer control outputs for MC3448A transceivers

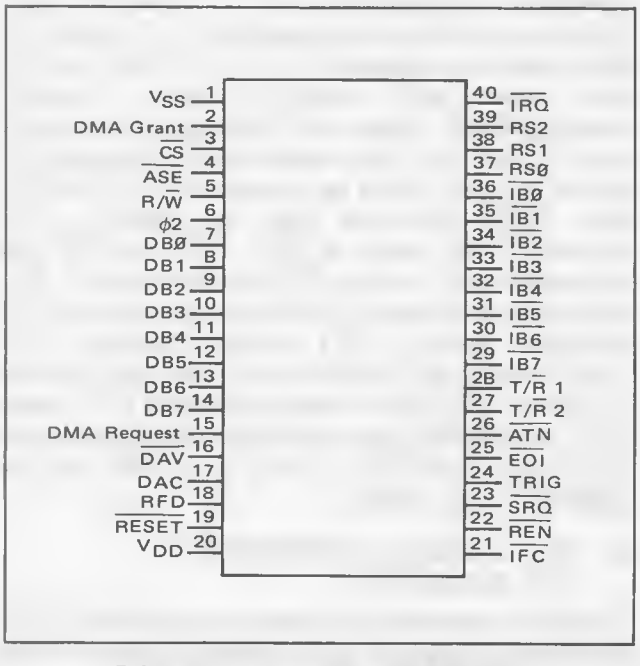


FIGURE 23 – Pinouts for MC68488 GPIA

NOTE: All discussions of pinouts and logic signals pertaining to the MC68488 or other M6800 family devices are with reference to positive logic (i.e. ≥ 2.4 V is logic high). This is directly opposite to the 488-1975 Standard. Thus MC68488 pin designations will be the logic inverse. Example: the NDAC line of 488-1975 is the DAC pin of the MC68488.

TABLE 10—Description of MC68488 Pinouts

Symbol	Description
DBØ–DB7	Allows Data Transfer Between the MPU and the MC68488
$\overline{IBØ}$–$\overline{IB7}$	Allows Data Transfer Between the MC68488 and the 488 Bus
$\overline{CS}$	Chip Select
RSØ–RS2	Register Selects
$\overline{IRQ}$	Interrupt Request
$\overline{RESET}$	Used to Initialize the Chip During Power-on
DMA GRANT	Direct Memory Access Grant
DMA REQUEST	Direct Memory Access Request
$\overline{ASE}$	Address Switch Enable
DAC	Data Accepted
RFD	Ready For Data
$\overline{DAV}$	Data Valid
$\overline{ATN}$	Attention
$\overline{IFC}$	Interface Clear
SRQ	Service Request
$\overline{REN}$	Remote Enable
$\overline{EOI}$	End Or Identify
T/$\overline{R}$1–2	Transmit/$\overline{\text{Receive}}$ Transceiver Control Outputs
φ2	Derivative of MPU φ2 Clock
R/$\overline{W}$	Read/$\overline{\text{Write}}$ Line From MPU
TRIG	Trigger out corresponds to GET and fget command
V_{SS}	Ground
V_{DD}	+5 V Power Supply

GPIA/MPU Interface Signals

The MC68488 interfaces to the MC6800 MPU with an eight-bit bidirectional data bus, a chip select, Read/$\overline{\text{Write}}$ line, Reset line, three register select lines, an interrupt request line, two DMA control lines, and an address switch enable line.

GPIA Bidirectional Data (DBØ-DB7)—The bidirectional data lines allow the transfer of data between the MPU and the GPIA. The data bus output drives are three state devices that remain in the high impedance (off) state except when the MPU performs a GPIA read operation. The Read/$\overline{\text{Write}}$ line is in the read state when the GPIA is selected for a read operation.

GPIA Chip Select ($\overline{\text{CS}}$)—This input signal is used to select the GPIA. $\overline{\text{CS}}$ must be low for selection of the device. Chip select decoding is normally accomplished with logic external to the chip.

GPIA Read/$\overline{\text{Write}}$ Line (R/$\overline{\text{W}}$)—This signal is generated by the MPU to control register access and direction of data transfer on the data bus. A low state on the GPIA Read/$\overline{\text{Write}}$ allows for the selection of one of seven write only registers when used in conjunction with the register select lines; RSØ, RS1, RS2. A high state on the GPIA Read/$\overline{\text{Write}}$ allows for the selection of one of eight read only registers when used in conjunction with register select lines RSØ, RS1, RS2.

GPIA Register Select (RSØ, RS1, RS2)—The three register select lines are used to select the various registers inside the GPIA. These three lines are used in conjunction with the Read/$\overline{\text{Write}}$ line to select a particular register that is to be written or read. Table 11 shows the register select coding.

Interrupt Request ($\overline{\text{IRQ}}$)—The $\overline{\text{IRQ}}$ output goes to the common interrupt bus for the MPU. This is an open drain output which is wire-ORed to the $\overline{\text{IRQ}}$ bus. The $\overline{\text{IRQ}}$ is set false (low) when an enabled interrupt occurs and stays false until the MPU reads from the interrupt status register.

$\overline{\text{Reset}}$—The active low $\overline{\text{Reset}}$ line is used to initialize the chip during power-on start up. $\overline{\text{Reset}}$ will be driven by an external power-up reset circuit.

DMA Control Lines (DMA Grant, DMA Request)—The DMA request line is used to signal waiting data when BI is set high for a DMA controller. The DMA request line is set high if either the BI or BO interrupt flags are set in the Interrupt Status Register (RØW). The DMA request line is cleared when the DMA grant is made true. The DMA grant line is used to signal the GPIA that the DMA controller has control of the MPU data and address lines. *DMA Grant must be grounded when not in use!*

Address Switch Enable ($\overline{\text{ASE}}$)—The $\overline{\text{ASE}}$ output is used to enable the device address switch three state buffers to allow the instrument address switch to be read on the MPU bus.

Clock Input (Clk) — The Clk input is normally a derivative of the MPU ϕ2 clock. It is designed to operate at clock rates up to 1 MHz.

MC68488-GPIA/488 Interface Bus Signals

The GPIA provides a set of eighteen interface signal lines between the M6800 and the IEEE Standard 488 bus.

Signal Lines ($\overline{\text{IBØ}}$-$\overline{\text{IB7}}$)—These bidirectional lines allow for the flow of seven bit ASCII interface messages and device dependent messages. Data appears on these lines in a bit-parallel byte-serial form. These lines are buffered by the MC3448A transceivers and applied to the 488 bus (DIO1-DIO8).

Byte Transfer Lines (DAC, RFD, $\overline{\text{DAV}}$)—These lines allow for proper transfer of each data byte on the bus between sources and acceptors. RFD goes passively true indicating that all acceptors are "ready for data." A source will indicate the "data is valid" by pulling $\overline{\text{DAV}}$ low. Upon the reception of valid data by all acceptors, DAC will go passively true indicating that the "data has been accepted" by all acceptors.

Bus Management Lines ($\overline{\text{ATN}}$, $\overline{\text{IFC}}$, $\overline{\text{SRQ}}$, $\overline{\text{EOI}}$, $\overline{\text{REN}}$)—These lines are used to manage an orderly flow of information across the interface lines.

Attention ($\overline{\text{ATN}}$)—Is sent true over the interface to disable current talkers and listeners freeing the signal lines ($\overline{\text{IBØ}}$-$\overline{\text{IB7}}$). During the $\overline{\text{ATN}}$ active state devices monitor the signal path for addressing or an interface command. Data flows on the signal lines when $\overline{\text{ATN}}$ is inactive (high).

Interface Clear ($\overline{\text{IFC}}$)—Is used to put the interface system into a known quiescent state.

Service Request ($\overline{\text{SRQ}}$)—Is used to indicate a need for attention in addition to requesting an interruption in the current sequence of events. This indicates to the controller that a device on the bus is in need of service.

Remote Enable ($\overline{\text{REN}}$)—is used to select one of two alternate sources of device programming data, local or remote control.

End or Identify ($\overline{\text{EOI}}$)—is used to signal the end of a multiple byte transfer sequence and in conjunction with $\overline{\text{ATN}}$ executes a parallel polling sequence.

Transmit/Receive Control Signals (T/$\overline{\text{R}}$ 1, T/$\overline{\text{R}}$ 2)—These two signals are used to control the quad transceivers which drive the interface bus. It is assumed that transceivers equivalent to the MC3448A will be used where each transceiver has a separate transmit/$\overline{\text{receive}}$ control pin. These pins can support one TTL load each. The outputs can then be grouped as shown in Figure 26 (in the section on the MC3448A) with $\overline{\text{SRQ}}$ hardwired high to transmit. The transmit/$\overline{\text{receive}}$ inputs of $\overline{\text{REN}}$, $\overline{\text{IFC}}$, and $\overline{\text{ATN}}$ are hardwired low to receive. $\overline{\text{EOI}}$ is controlled by T/$\overline{\text{R}}$ 1 through the MC3448A (or an equivalent) allowing it to transmit or receive. T/$\overline{\text{R}}$ 1 operates exactly as T/$\overline{\text{R}}$ 2 except during the parallel polling sequence. During parallel poll $\overline{\text{EOI}}$ will be made an input by T/$\overline{\text{R}}$ 1 while $\overline{\text{DAV}}$ and $\overline{\text{IBØ}}$/$\overline{\text{IB7}}$ lines are outputs. During Serial Poll T/$\overline{\text{R}}$ 1 will make $\overline{\text{EOI}}$ an input with $\overline{\text{DAV}}$ and the $\overline{\text{IBØ}}$/$\overline{\text{IB7}}$ bus as outputs.

B. GPIA INTERNAL CONTROLS AND REGISTERS*

***NOTE:** Upper and lower case type designations will be used with the register bits to indicate remote or local messages respectively.

There are fifteen locations accessible to the MPU data bus which are used for transferring data to control the various functions on the chip and provide current chip status. Seven of these registers are write only and eight registers are read only. The various registers are accessed according to the three least significant bits of the MPU address bus and the status of the Read/$\overline{\text{Write}}$ line. Table 11 shows proper coding for register access within the MC68488. One of the fifteen registers is external to the IC but an address switch register is provided for reading the address switches. Figure 24 shows actual bit contents of each of the registers.

Data-In Register R7R — The data-in register is an actual eight-bit storage register used to move data from the interface bus when the chip is a listener. Reading the register does not destroy information in the data-out register. Normally DAC (data accepted) will remain low until the MPU removes the byte from the data-in register. The chip will automatically finish the handshake by allowing DAC to go high. In RFD (ready for data) holdoff mode, a new handshake is not initiated until a command is sent allowing the chip to release holdoff. This will delay a talker until the available information has been processed.

Data-Out Register R7W — The data-out register is an actual eight-bit storage register used to move data out of the chip onto the interface bus. Reading from the data-in register has no effect on the information in the data-out register. Writing to the data-out register has no effect on the information in the data-in register.

TABLE 11 — Register Access

RS2	RS1	RSØ	R/$\overline{\text{W}}$	Register Title	Register Symbol
0	0	0	1	Interrupt Status	RØR
0	0	0	0	Interrupt Mask	RØW
0	0	1	1	Command Status	R1R
0	0	1	0	Unused	—
0	1	0	1	Address Status	R2R
0	1	0	0	Address Mode	R2W
0	1	1	1	Auxiliary Command	R3R
0	1	1	0	Auxiliary Command	R3W
1	0	0	1	Address Switch*	R4R
1	0	0	0	Address	R4W
1	0	1	1	Serial Poll	R5R
1	0	1	0	Serial Poll	R5W
1	1	0	1	Command Pass-Through	R6R
1	1	0	0	Parallel Poll	R6W
1	1	1	1	Data In	R7R
1	1	1	0	Data Out	R7W

*External to MC68488.

Interrupt Mask Register RØW — The Interrupt Mask Register is a seven-bit storage register used to select the particular events that will cause an interrupt to be sent to the MPU. The seven control bits may be set independently of each other. If dsel (bit 7 of the Address Mode Register) is set high CMD bit 2 will interrupt on SPAS or RLC. If dsel is set low CMD will interrupt on UACG, UUCG, and DCAS in addition to RLC and SPAS. The Command Status Register R1R may then be used to determine which command caused the interrupt. Setting GET bit 5 allows an interrupt to occur on Group Execute Trigger Command. END bit 1 allows an interrupt to occur if $\overline{\text{EOI}}$ is true (low) and $\overline{\text{ATN}}$ is false (high). APT bit 3 allows an interrupt to occur indicating that a secondary address is available to be examined by the MPU if apte (bit Ø of Address Mode Register) is enabled and listener or talker primary address is received and a Secondary Command Group is received. A typical response for a valid secondary address would be to set msa (bit 3 of Auxiliary Command Register) true and dacr (bit 4 Auxiliary Command Register) true, releasing the DAC handshake. BI indicates that a data byte is waiting in the data-in register. Bl is set high when data-in register is full. BO indicates that a byte from the data-out register has been accepted. BO is set when the data-out register is empty. $\overline{\text{IRQ}}$ allows any interrupt to be passed to the MPU.

The Interrupt Status Register RØR — The Interrupt Status Register is a seven-bit storage register which corresponds to the Interrupt Mask Register with an additional bit INT bit 7. Except for the INT bit the other bits in the status register are set regardless of the state of the interrupt mode register when the corresponding event occurs. The $\overline{\text{IRQ}}$ (MPU interrupt) is cleared when the MPU reads from the register. INT bit 7 is the logical OR of the other six bits ANDed with the respective bit of RØW.

Serial Poll Register R5R/W — The Serial Poll Register is an eight-bit storage register which can be both written into and read by the MPU. It is used for establishing the status byte that the chip sends out when it is serial poll enabled. Status may be placed in bits Ø through 5 and bit 7. Bit 6 rsv (request for service) is used to drive the logic which controls the $\overline{\text{SRQ}}$ line on the bus telling the controller that service is needed. This same logic generated the signal SRQS which is substituted in bit 6 position when the status byte is read by the MPU $\overline{\text{IBØ}}$-$\overline{\text{IB7}}$. In order to initiate a rsv (request for service), the MPU sets bit 6 true (generating rsv signal) and this in turn causes the chip to pull down the $\overline{\text{SRQ}}$ line. SRQS is the same as rsv when SPAS is false. Bit 6 as read by the MPU will be the SRQS (Service Request State).

Parallel Poll Register R6W — This register will be loaded by the MPU and the complement of the bits in this register will be delivered to the instrument bus $\overline{\text{IBØ}}$-$\overline{\text{IB7}}$ during PPAS (Parallel Poll Active State). This register powers up in the PPØ (Parallel Poll No Capability) state. The reset bit (R3R/W) will clear this register to the PPØ state.

The parallel poll interface function is executed by

this chip using the PP2 subset (Omit Controller Configuration Capability). The controller cannot directly configure the parallel poll output of this chip. This must be done by the MPU. The controller will be able to indirectly configure the parallel poll by issuing an addressed command which has been defined in the MPU software.

Address Mode Register R2W — The address mode register is a storage register with six bits for control: to, lo, hlde, hlda, dsel, and apte. The to bit 6 selects the talker/listener and addresses the chip to talk only. The lo bit 5 selects the talker/listener and sets the chip to listen only. The apte bit Ø is used to enable the extended addressing mode. If apte is set low the device goes from the TPAS (Talker Primary Address State) directly to the TADS (Talker Addressed State). The hlda bit 2 holds off RFD (Ready for Data) on ALL DATA until rfdr is set true. The hlde bit 3 holds off RFD on $\overline{EOI}$ enabled (low) and $\overline{ATN}$ not enabled (high). This allows the last byte in a block of data to be continually read as needed. Writing rfdr true (high) will release the handshake.

Address Status Register R2R — The address status register is not a storage register but simply an eight-bit port used to couple internal signal nodes to the MPU bus. The status flags represented here are stored internally in the logic of the chip. These status bits indicate the addressed state of the talker/listener as well as flags that specify whether the chip is in the talk only or listen only mode. The ATN, bit 4, contains the condition of the Attention Line. The ma signal is true when the chip is in:

TACS — Talker Active State
TADS — Talker Addressed State
LACS — Listener Active State
LADS — Listener Addressed State
SPAS — Serial Poll Active State

Address Switch Register R4R — The address switch register is external to the chip. There is an enable line ($\overline{ASE}$) to be used to enable three-state drivers connected between the address switches and the MPU. When the MPU addresses the address switch register the enable line directs the switch information to be sent to the MPU. The five least significant bits of the eight-bit register are used to specify the bus address of the device and the remaining three bits may be used at the discretion of the user. The most probable use of one or two of the bits is for controlling the listener only or talk only functions. (See Figures 25 and 29 and the section on Addressing.)

Address Register R4W — The Address Register is an eight-bit storage register. The purpose of this register is to carry the primary address of the device. The primary address is placed in the five least significant bits of the register. If external switches are used for device addressing these are normally read from the Address Switch Register and then placed in the Address Register by the MPU.

AD1 through AD5 bits Ø–5 are for the device's address. The lsbe bit 7 is set to enable the Dual Primary Addressing Mode. During this mode the device will respond to two consecutive addresses, one address with AD1 equal to Ø and the other address with AD1 equal to 1. For example, if the device's address is $ØF, the Dual Primary Addressing Mode would allow the device to be addressed at both $ØF and $ØE. The dal bit 6 is set to disable the listener and the dat bit 5 is set to disable the talker.

This register is cleared by the $\overline{Reset}$ input only (not by the reset bit of R3R/W).

When $\overline{ATN}$ is enabled and the primary address is received on the $\overline{IBØ–7}$ lines, the MC68488 will set bit 7 of the address status register (ma). This places the MC68488 in the TPAS or LPAS.

When $\overline{ATN}$ is disabled the GPIA may go to one of three states: TACS, LACS or SPAS.

Auxiliary Command Register R3R/W — Bit 7, reset, initializes the chip to the following states (reset is set true by external $\overline{Reset}$ input pin and by writing into the register from the MPU):

SIDS—Source Idle State
AIDS—Acceptor Idle State
TIDS—Talker Idle State
LIDS—Listener Idle State
LOCS—Local State
NPRS—Negative Poll Response State
PPIS—Parallel Poll Idle State
PUCS—Parallel Poll Unaddressed to Configure State
PPØ—Parallel Poll No Capability

rfdr (release RFD handshake) bit 6 allows for completion of the handshake that was stopped by RFD (Ready For Data) holdoff commands hlda and hlde.

fget (force group execute trigger) bit Ø has the same effect as the GET (Group Execute Trigger) command from the controller.

rtl (return to local) bit 2 allows the device to respond to local controls and the associated device functions are operative.

dacr (release DAC handshake) bit 4 is set high to indicate that the MPU has examined a secondary address or an undefined command.

ulpa (upper/lower primary address) bit 1 will indicate the state of the LSB of the address received on the DIO1–8 lines at the time the last Primary Address was received. This bit can be read but not written by the MPU.

msa (valid secondary address) bit 3 is set true (high) when TPAS (Talker Primary Addressed State) or LPAS (Listener Primary Addressed State) is true. The chip will become addressed to listen or talk. The primary address must have been previously received.

RFD, $\overline{DAV}$, DAC - (Ready For Data, Data Valid, Data Accepted). These bits assume the same state as the corresponding signal on the MC68488 package pins. The MPU may only read these bits.

dacd (data accept disable) bit 1 set high by the MPU

INTERRUPT STATUS REGISTER

(Read Only)

INT	BO	GET	X	APT	CMD	END	BI

INT — Logical OR of all other bits in this register ANDed with the respective bits in the interrupt mask register.
BO — A byte of data has been output
GET — A Group Execute Trigger has occurred
APT — An Address Pass-Through has occurred
CMD — SPAS + RLC + $\overline{dsel}$ (DCAS + UUCG + UACG) has occurred
END — An EOI has occurred with $\overline{ATN}$ = Ø
BI — A byte has been received

INTERRUPT MASK REGISTER

(Write Only)

IRQ	BO	GET	X	APT	CMD	END	BI

IRQ — Mask bit for IRQ pin
BO — Interrupt on byte output
GET — Interrupt on Group Execute Trigger
APT — Interrupt on Secondary Address Pass-Through
CMD — Interrupt on SPAS + RLC + $\overline{dsel}$ (DCAS + UUCG + UACG)
END — Interrupt on EOI and $\overline{ATN}$
BI — Interrupt on byte input

COMMAND STATUS REGISTER

(Read)

UACG	REM	LOK	X	RLC	SPAS	DCAS	UUCG

UACG — Undefined Addressed Command
REM — Remote Enabled
LOK — Local Lockout Enabled
RLC — Remote/Local State Changed
SPAS — Serial Poll Active State is in effect
DCAS — Device Clear Active State is in effect
UUCG — Undefined Universal Command

ADDRESS STATUS REGISTER

(Read Only)

ma	to	lo	ATN	TACS	LACS	LPAS	TPAS

ma — my address has occurred
to — the talk-only mode is enabled
lo — the listen-only mode is enabled
ATN — The Attention command is asserted
TACS — GPIA is in the Talker Active State
LACS — GPIA is in the Listener Active State
LPAS — GPIA is in the Listener Primary Addressed State
TPAS — GPIA is in the Talker Primary Addressed State

ADDRESS MODE REGISTER

(Write Only)

dsel	to	lo	X	hdle	hdla	X	apte

dsel — configure for automatic completion of handshake sequence on occurrence of GET, UACG, UUCG, SDC, or DCL commands
to — set to talk-only mode
lo — set to listen-only mode
hdle — Hold-off RFD on end
hdla — Hold-off RFD on all data
apte — Enable the address pass-through feature

ADDRESS SWITCH REGISTER

(Read Only)

UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1

AD1–AD5 — Device address
UD1–UD3 — User definable bits
When this "register" is addressed, the $\overline{ASE}$ pin is set which allows external address switch information from bus device to be read.

AUXILIARY COMMAND REGISTER

reset	rfdr	feoi	dacr	msa	rtl	dacd	fget	Write
	DAC	$\overline{DAV}$	RFE			ulpa		Read

reset — initialize the chip to the following status:
(1) all interrupts cleared
(2) following bus states are in effect: SIDS, AIDS, TIDS, LIDS, LOCS, PPIS, PUCS, and PPØ
(3) bit is set by $\overline{Reset}$ input pin
msa — if GPIA is in LPAS or TDAS, setting msa will force GPIA to LADS or TADS
rtl — return to local if local lockout is disabled
ulpa — state of LSB of bus at last-primary-address receive time
fget — force group execute trigger command from the MPU has occurred
rfdr — complete handshake stopped by RFD holdoff
feoi — set EOI true, clears after next byte transmitted
dacr — MPU has examined an undefined command or secondary address
dacd — prevents automatic handshake on Addresses or Commands

ADDRESS REGISTER

(Write Only)

lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1

lsbe — enable dual primary addressing mode
dal — disable the listener
dat — disable the talker
AD1–AD5 — Primary device address, usually read from address switch register
Register is cleared by the $\overline{Reset}$ input pin only.

SERIAL POLL REGISTER

(Read)

S8	SRQS	S6	S5	S4	S3	S2	S1

S1–S8 — Status bits
SRQS — Bus in Service Request State

SERIAL PDLL REGISTER

(Write)

S8	rsv	S6	S5	S4	S3	S2	S1

S1–S8 — Status bits
rsv — generate a service request

COMMAND PASS-THROUGH REGISTER

(Read Only)

B7	B6	B5	B4	B3	B2	B1	BØ

An eight-bit input port used to pass commands and secondary addresses to MPU which are not automatically interpreted by the GPIA

PARALLEL POLL REGISTER

(Write Only)

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

Bits delivered to bus during Parallel-Poll Active State (PPAS)
Register powers up in the PPØ state
Parallel Poll is executed using the PP2 subset

DATA-IN REGISTER

(Read Only)

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DIØ

DIØ–DI7 — Correspond to DID1-DID8 of the 488-1975 Standard and $\overline{IBØ}$–$\overline{IB7}$ of the MC6848B

DATA OUT REGISTER

(Write Only)

DO7	DO6	DO5	DD4	DO3	DO2	DD1	DOØ

DOØ–DO7 — Correspond to DIO1–DIO8 of the 488-1975 Standard and $\overline{IBØ}$–$\overline{IB7}$ of the MC68488

FIGURE 24 — Bit Contents of Registers

will prevent automatic handshake on Addresses or Commands. dacr is used to release the handshake.

feoi (forced end or identify) bit 5 tells the chip to send $\overline{EOI}$ low with the next data byte transmitted. The $\overline{EOI}$ line is then returned high after the next byte is transmitted. NOTE: The following signals are not stored but revert to a false (low) level one clock cycle (MPU ϕ2) after they are set true (high):

rfdr
feoi
dacr

These signals can be written but not read by the MPU.

Command Status Register R1R — The command status register flags commands or states as they occur. These flags or states are simply coupled onto the MPU bus from internal storage nodes. There are five major address commands. REM shows the remote/local state of the talker/listener. REM bit 6 set low implies the local state. LOK bit 5 shows the local lockout status of the talker/listener. RLC bit 3 is set when a change of state of the remote/local flip-flop occurs and reset when the command status register is read. DCAS bit 1 indicates that either the device clear or selected device clear has been received activating the device clear function. SPAS bit 2 indicates that the SPE command has been received activating the device serial poll function. UACG bit 7 indicates that an undefined address command has been received and depending on programming the MPU decides whether to execute or ignore it. UUCG bit Ø indicates that an undefined universal command has been received.

Command Pass-Through Register R6R — The command pass through is an eight-bit port with no storage. When this port is addressed by MPU it connects the instrument data bus ($\overline{IB0}$–$\overline{IB7}$) to the MPU data bus DBØ–DB7. This port can be used to pass commands and secondary addresses that aren't automatically interpreted through to the MPU for inspection.

C. ADDRESSING

The MC68488 has internal logic which can recognize certain bit patterns on the $\overline{IB0}$–$\overline{IB7}$ lines. When ATN is true, this logic must determine if the bits comprise a valid command or an address. Talk and Listen addresses must have the X1ØXXXXX and XØ1XXXXX bit patterns respectively. Multiline commands have XØØXXXXX or X11XXXXX formats. In addition, if a Universal Command is not one that is recognized by the MC68488 logic, the UUCG bit in the Command Status Register will be set. See Table 12 for command patterns, and Table 7 for defined commands.

The GPIA has the ability to respond to two addresses in the Dual Primary Addressing mode. In this mode, the device will respond to two addresses which differ only in the LSB (AD1 of the address register) of address. To determine which of the two was actually transmitted, the contents of the ulpa bit of the Auxiliary Command Register must be examined.

The use of the alphanumeric portion of the ISØ–7 code (commonly ASCII) is strongly recommended for coding of device dependent messages and data. Thus, the defined Universal and Addressed Commands can be coded as either ASCII or equivalent hexadecimal characters, where the MSB will be assumed Ø since it is always a "don't care" anyway. Table 13 shows the commands and addresses and their ASCII and hexadecimal equivalents.

TABLE 12 — Command & Address Formats

	$\overline{IB7}$ . . . $\overline{IB0}$
Addressed Commands	XØØØXXXX
Universal Commands	XØØ1XXXX
Listen Addresses	XØ1XXXXX
Talk Addresses	X1ØXXXXX
Secondary Commands	X11XXXXX

TABLE 13 — ASCII and Hexadecimal Equivalents

ATN = 1

Command Mnemonic	ASCII Code	Hex Equivalent ($)
GTL	SOH	Ø1
SDS	EOT	Ø4
PPC	ENQ	Ø5
GET	BX	Ø8
TCT	HT	Ø9
LLO	DC1	11
DCL	OC4	14
PPU	NAK	15
SPE	CAN	18
SPD	EM	19
Secondary Commands	Blank thru ~	6Ø thru 7E

ATN = 1

ADDRESS	ASCII CODE	Hex Equivalent ($)
Listen Address	SP thru >	2Ø thru 3E
Unlisten	?	3F
Talk Address	@ thru ^	4Ø thru 5E
Untalk	—	5F

Device Addressing Procedure:

1. Set Address Switches to desired Device Address
2. Turn On power
3. Program MPU to Address Register R4R (this Enables the Three-State Address Switch buffers and applies the contents to the MPU Data Bus). Load contents into Accumulator A.

4. Store contents of Accumulator A into Register R4W. This is the same address as in the previous step. Only the state of R/$\overline{W}$ has changed.
5. Pull $\overline{RESET}$ pin LOW to initialize all internal registers (except R4W).
6. After step 5 is completed, at any time the device's Primary Address is presented on the $\overline{IB}\emptyset$–$\overline{7}$ lines and $\overline{ATN}$ is Enabled, the GPIA will recognize its address. This will cause the GPIA to set bit 7 of the Address Status Register and place the GPIA in either the LPAS or TPAS State.

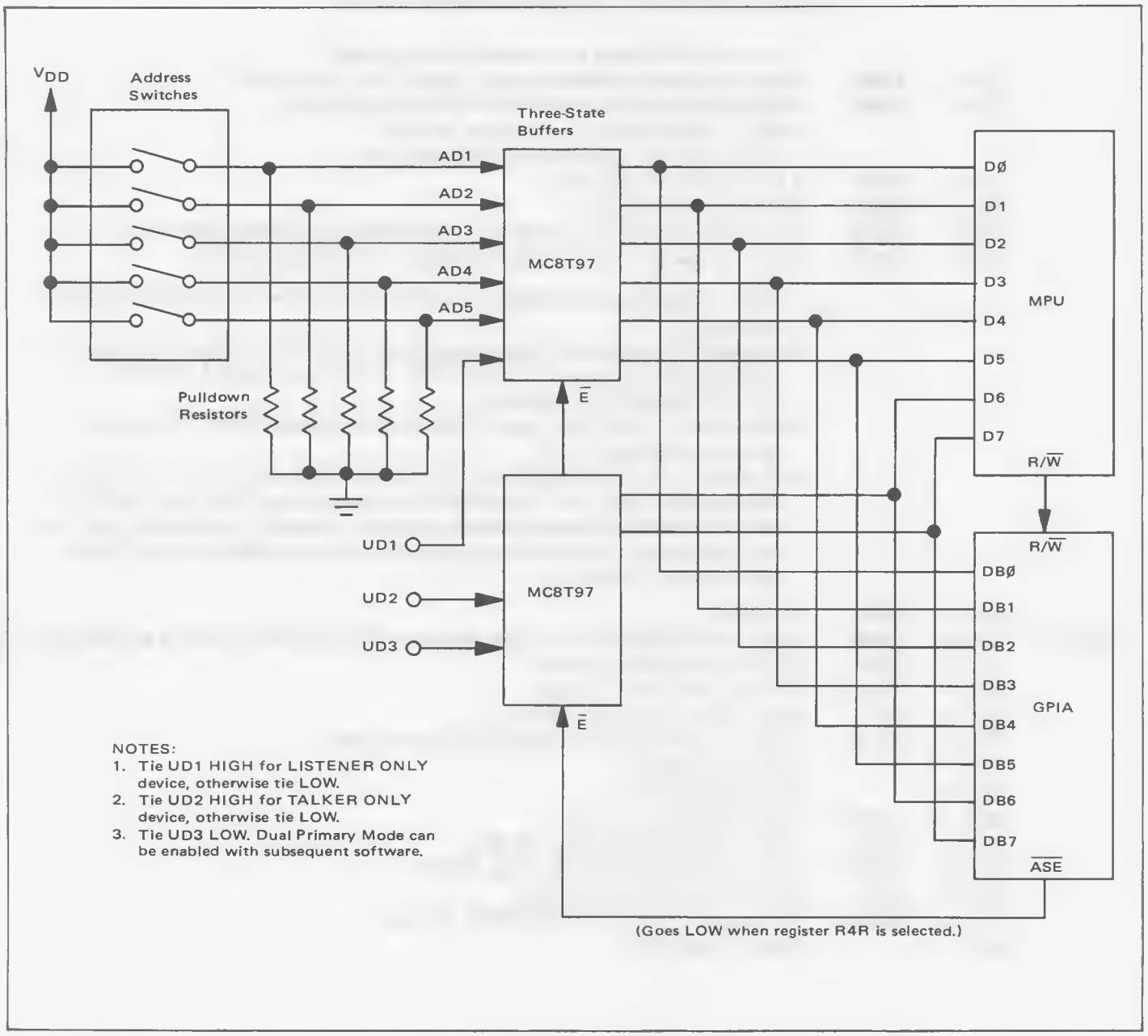


FIGURE 25 — Device Address Connections

D. PROGRAMMING EXAMPLE

Programming of the MC6800/68488 is beyond the scope of this brochure. Techniques of programming are discussed in the *M6800 Microprocessor Programming Reference Manual.*

An example of a GPIB program implementation follows. This program is for illustrative purposes only and is not intended to necessarily be the most efficient or practical software for any given application.

BASIC SOFTWARE TALKER CONFIGURATION

```
                                  Assume the MC68488 is at address location $5000.
        LDAA    $5004             READ the device's address on the ADDRESS SWITCHES.
        STAA    $5004             WRITE the address into the ADDRESS REGISTER.
                                  NOTE: Assume the device's address was $0A.
                                    (AD5-AD1 corresponds to 01010 respectively.)
        LDAA    #$00              LOAD ACC A with zeros.
        STAA    $5003             This clears the reset bit.
        STAA    $5000             Mask all interrupts (if desired) in the INTERRUPT MASK REGISTER.
        STAA    $5002             Select no special features in the ADDRESS MODE REGISTER.

                                  NOTE: At this time the controller will address the device to TALK in the following
                                    manner.
                                  ENABLE ATN and send mta (my talk address) on the DIO1-8 lines which would be
                                    X1001010 ($4A). Now DISABLE ATN. A READ of $5002 ADDRESS
                                    STATUS REGISTER will show:
                                  $89 ma (Bit 7), TACS (Bit 3), and TPAS (Bit 0) will be set HIGH. At this time the
                                    device is ready to TALK.
                                  BO (Bit 6) of the INTERRUPT STATUS REGISTER will be HIGH. Writing a
                                    byte to $5007 DATA OUT REGISTER will reset BO to ZERO. BO is set HIGH
                                    when the device(s) listening accepts the data. A possible software approach to
                                    output five bytes of data from memory pointed to by the INDEX REGISTER to
                                    the 488 bus is as follows:
        LDAB    #$04              Counter.
LOOP    LDAA    $5000             LOAD ACCUMULATOR A with contents of INTERRUPT STATUS REGISTER.
        CMPA    #$50              Check for Bit (BO) to be set.
        BEQ     LOOP              If BO is Zero, keep checking.
        LDAA    0,X               Get a byte of data (BO went set).
        STAA    $5007             Output to DATA OUT REGISTER and on to bus.
        INX                       Increment pointer.
        DECB                      Decrement counter.
        BNE     LOOP              If not finished, loop back.
        LDAA    #$20              Load ACCUMULATOR with 20.
        STAA    $5003             SET EOI (this precedes last byte of data).
        LDAA    0,X               Remove last byte from buffer.
        STAA    $5007             Writes to DATA OUT REGISTER and bus.
        RTS                       End of Subroutine.
```

BASIC SOFTWARE LISTENER CONFIGURATION

```
                                  Assume the MC68488 is at address location $5ØØØ.
        LDAA    $5ØØ4             READ the device's address on the ADDRESS SWITCHES.
        STAA    $5ØØ4             WRITE the address into the ADDRESS REGISTER.
                                  NOTE: Assume the device's address was $Ø6.
                                    (AD5-AD1 corresponds to ØØ11Ø respectively.)
        LDAA    #$ØØ              LOAD ACC A with zeros.
        STAA    $5ØØ3             This clears the reset bit.
        STAA    $5ØØØ             Mask all interrupts (if desired) in the INTERRUPT MASK REGISTER.
        STAA    $5ØØ2             Select no special features in the ADDRESS MODE REGISTER.

                                  NOTE: At this time the controller will address the device to LISTEN in the
                                    following manner:
                                  ENABLE ATN and send mla (my listen address) on the DIO1-8 lines which would
                                    be XØ1ØØ11Ø ($26). Now DISABLE ATN. A READ of $5ØØ2 ADDRESS
                                    STATUS REGISTER will show $86 ma (Bit 7), LACS (Bit 2), and LPAS (Bit 1)
                                    will be set HIGH. At this time the device is ready to LISTEN.
                                  BI (Bit Ø) of the INTERRUPT STATUS REGISTER will be LOW. BI will go
                                    HIGH to indicate that a data byte is available in the DATA-IN REGISTER at
                                    $5ØØ7. Reading the DATA-IN REGISTER will reset BI (Bit Ø). A possible
                                    software approach could be as follows: Accept data from the 488 bus to a memory
                                    buffer pointed to by INDEX REGISTER.
LOOP 1  LDAA    $5ØØØ             Load ACC A with contents of INTERRUPT STATUS REGISTER.
        TAP                       Transfers ACC A contents to CONDITION CODE REGISTER.
        BCC     LOOP 1            LOOP until carry bit is set. This indicates BI is set in RØR.
        BVS     LOOP 2            BRANCH to LOOP 2 if overflow is set, indicating END, bit 1,
                                    of RØR has set (i.e., Controller has sent EOI).
        LDAA    $5ØØ7             LOAD DATA-IN REGISTER into ACC A. This resets bit BI.
        STAA    Ø,X               STORE the data byte in the buffer.
        INX                       Increment pointer.
        BRA     LOOP 1            BRANCH back to LOOP 1 and check to see if BI is set.
LOOP 2  INX                       Increment pointer.
        LDAA    $5ØØ7             Get the last byte of data from the DATA-IN REGISTER.
        STAA    Ø,X               Put last byte in the buffer.
        RTS                       End of Subroutine.
```

EXAMPLE ADDRESS MAP

Hexadecimal Address	MC68488 Registers (R/$\overline{W}$)
$5ØØØ	Interrupt Status/Interrupt Mask
$5ØØ1	Command Status/—
$5ØØ2	Address Status/Address Mode
$5ØØ3	Auxiliary Command/Auxiliary Command
$5ØØ4	Address Switch/Address
$5ØØ5	Serial Poll/Serial Poll
$5ØØ6	Command Pass-thru/Parallel Poll
$5ØØ7	Data In/Data Out

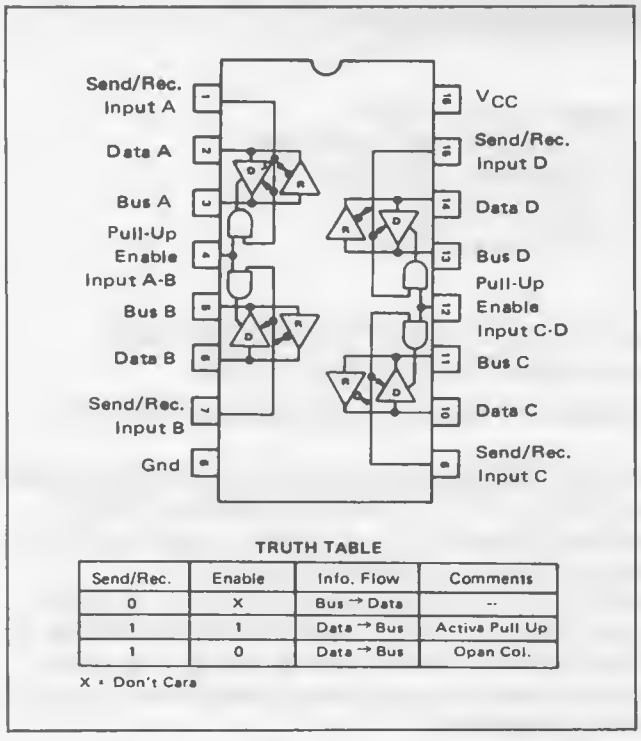
V. MC3448A BUS TRANSCEIVERS

The MC3448A is a quad bidirectional transceiver for mating MOS or bipolar logic systems to the 488 bus. Each channel provides back-to-back driver and receiver elements plus the required bus terminations. Direction of data flow is controlled by three-state disabling of the undesired direction element (i.e. driver or receiver). Schottky technology assures high speed while PNP buffered input structures guarantee low input loading for MOS compatibility. Both driver and receiver elements are non-inverting.

A pullup enable input is provided on each pair of drivers which allows selection of open-collector or three-state driver configuration.

Additional features include:

- Minimum receiver hysteresis of 400 mV for improved noise immunity
- Power up/down protection to assure that no invalid information is transmitted to the bus during these time periods
- No bus loading (including terminations) when power is removed from the device
- Fast propagation delay times
- Selection of three-state or open-collector configurations.



TRUTH TABLE

Send/Rec.	Enable	Info. Flow	Comments
0	X	Bus → Data	--
1	1	Data → Bus	Active Pull Up
1	0	Data → Bus	Open Col.

X = Don't Care

FIGURE 26 – Pinouts and Truth Table

The pinouts and truth table for the MC3448A are given in Figure 26.

Four MC3448A transceivers are required to buffer the 16 bus lines. Figure 27 shows the connections of the transceiver Send/Receive and Pullup Enable inputs.

Figure 28 shows the proposed Motorola 488-1975 system along with the required interconnections, while Figure 29 shows an expanded system with bus extenders, and provisions for generous amounts of ROM, RAM, peripherals, and I/O.

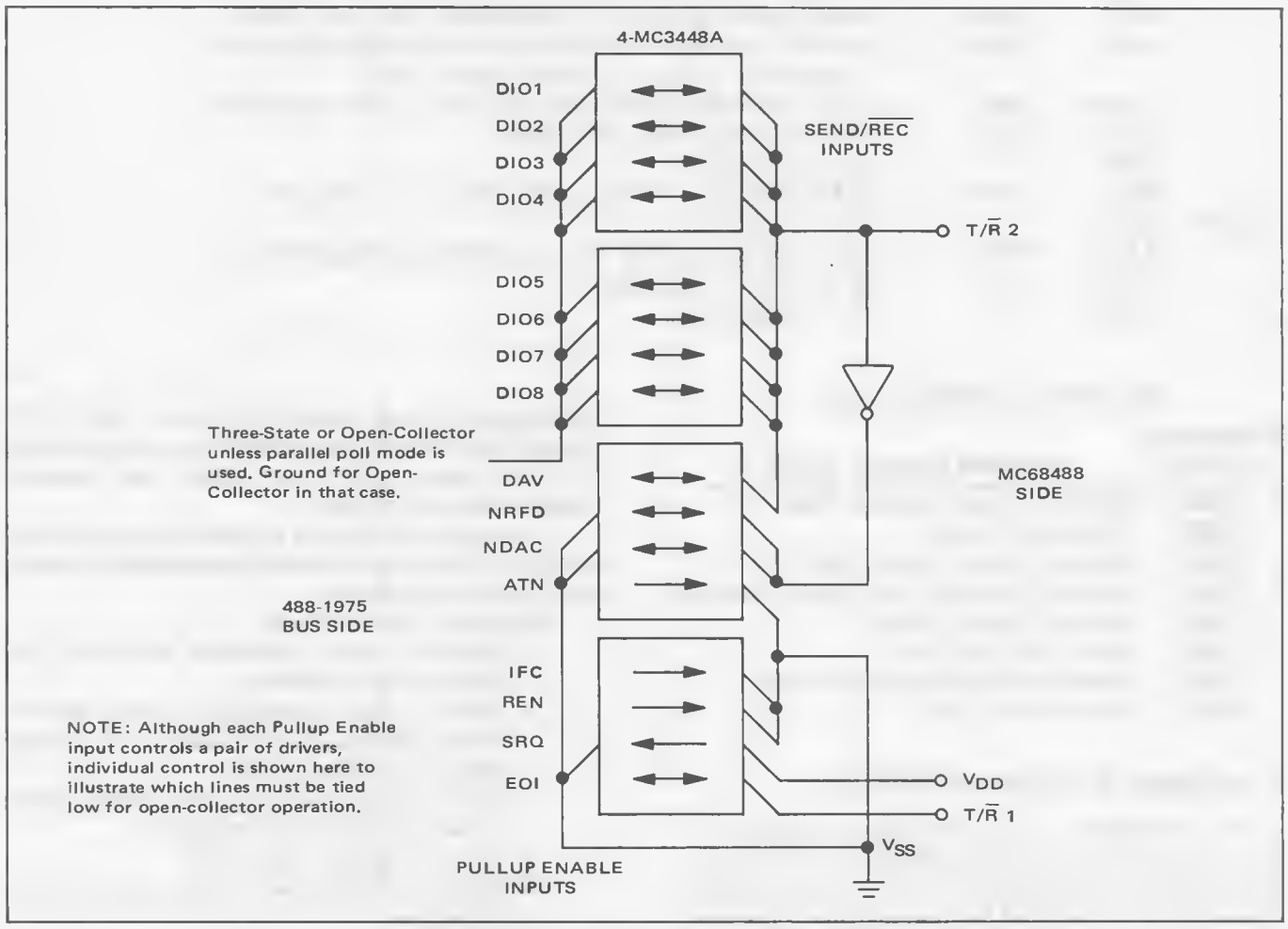
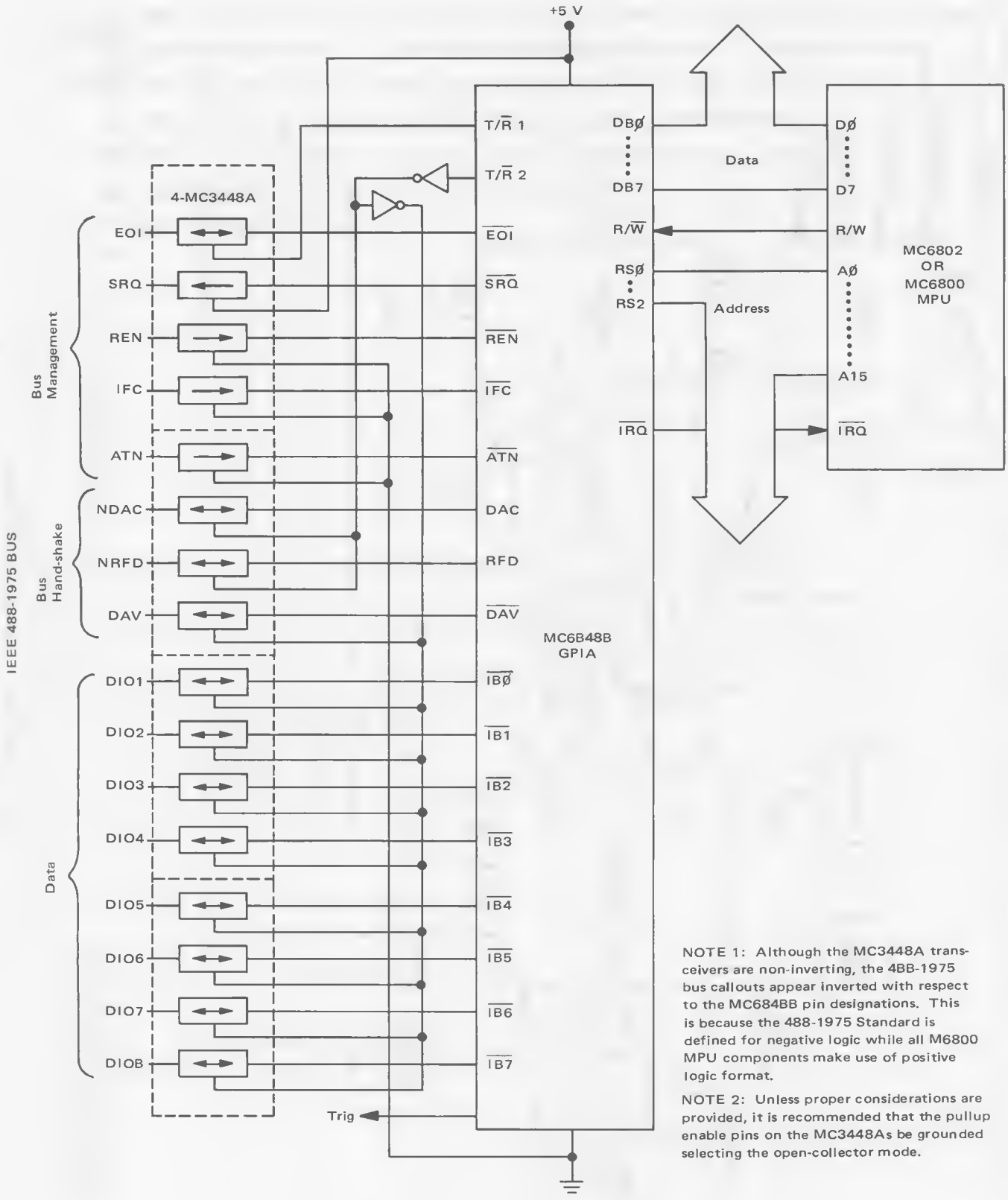


FIGURE 27 – MC3448A Direction Control and Pullup Enable



NOTE 1: Although the MC3448A transceivers are non-inverting, the 4BB-1975 bus callouts appear inverted with respect to the MC684BB pin designations. This is because the 488-1975 Standard is defined for negative logic while all M6800 MPU components make use of positive logic format.

NOTE 2: Unless proper considerations are provided, it is recommended that the pullup enable pins on the MC3448As be grounded selecting the open-collector mode.

FIGURE 28 — Simple System Configuration

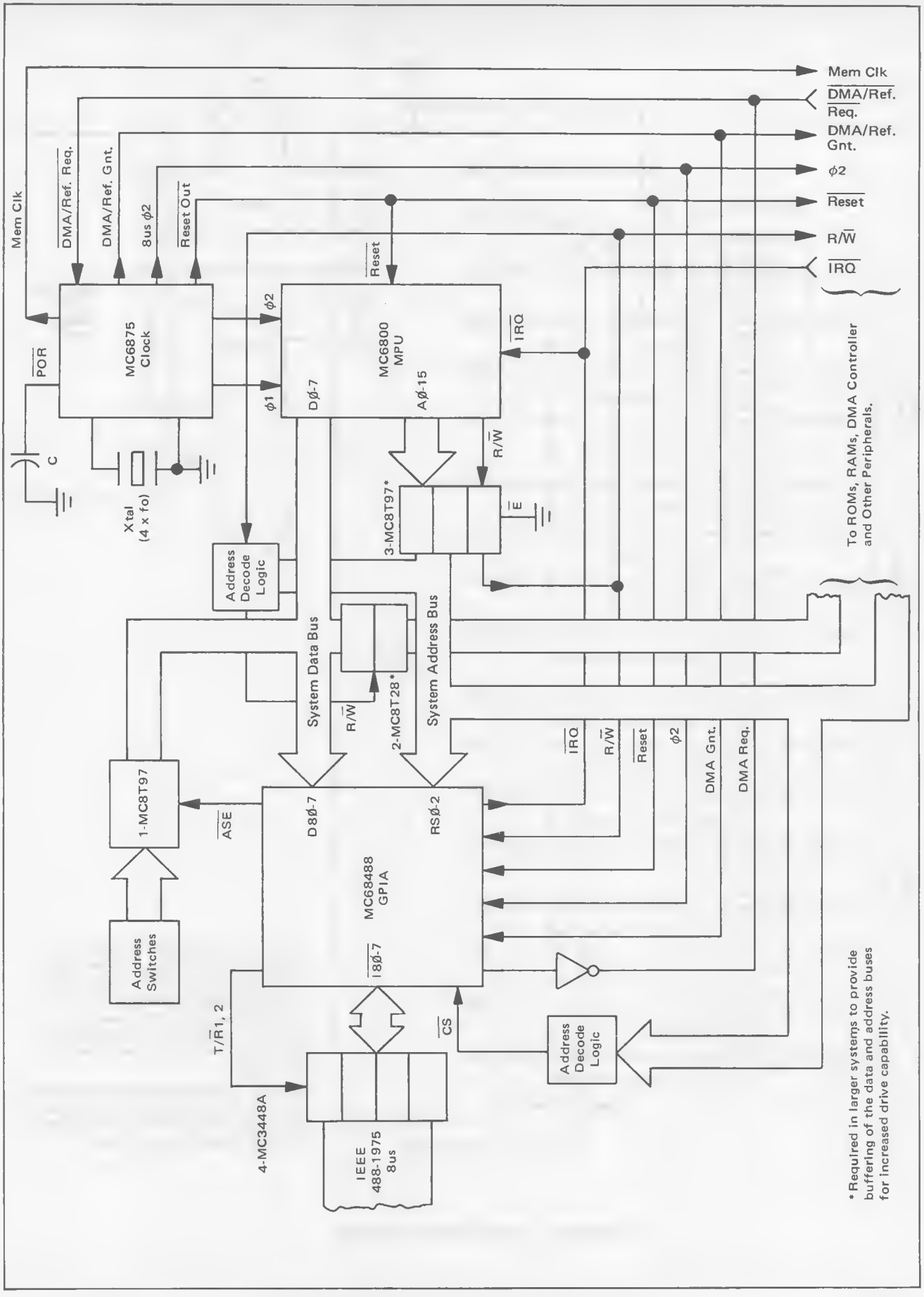


FIGURE 29 – Expanded GPIA/MPU System

CONCLUSION

The Motorola MC6800/MC68488/MC3448A is a cost-effective and highly flexible solution to the implementation of the IEEE 488-1975 instrumentation standard. This chip complement provides complete talker/listener capabilities. Controller functions may also be generated with these ICs and some supportive logic and memory/software.

In addition to the two dedicated 488-1975 devices, the entire M6800 family is available to configure more extensive measurement system designs. ROMs, RAMs, EPROMs, PIAs, ACIAs, Clocks, Programmable Timers, Bus Extenders and Modems are also offered in the M6800 family of devices.

System/Software development is simplified with support ranging from extensive technical literature and MPU training classes to the EXORcisor development system with a full range of options including floppy disk storage, and CRT display. For smaller system development, the $235.00 (1–5 quantity) D2 Microcomputer design kit provides keyboard, display and audio cassette memory interface at a minimum expense.

EXORcisor is a trademark of Motorola Inc.